Pure Data Workshop
Overview

Venue: Room M210
Instructor: Juan Ignacio Mendoza Garay, Doctoral Student, University of Jyväskylä

SESSION 1 (November 23, 2017, 10:15 to 14:00*):

- Demonstration of patches
- Introduction and similarities with MAX/MSP/Jitter
- Individual work: Mini-project

  Ideas for the Mini-project:
  - Translate a patch for MAX/MSP/Jitter to Pure Data
  - Do a part of your final project (so your final project will be in Pure Data)

ASSIGNMENT 6: First version of mini-project and presentation.

SESSION 2 (November 30, 2017, 10:15 to 14:00*):

- Students present their first version of the mini-project to the class. Discussion, ideas and suggestions for improvement.
- Individual work: Revision of mini-project.

ASSIGNMENT 7: Final version of mini-project

* There will be a lunch-break from 11:30 to 12:15

Pure Data Workshop

Introduction and similarities with MAX/MSP/Jitter

A patch tested in Pd-Extended 0.42.5 will be used to demonstrate the following:

- Choosing and installing a Pd version (https://puredata.info/downloads)
- Window navigation and documentation.
- Basic non-audio objects and connections.
- Send, receive and variables.
- Storing data.
- Networking
- Audio
- Video (Gem)
- Human Interfacing Devices
- Making a Graphical User Interface (GUI)
- Making abstractions and a self-contained application

Pure Data Workshop
Mini-Project

REQUIREMENTS:

A. An original project consisting in a Pure Data patch that does something useful, appealing to the senses or both. The project may include abstractions and other files required by the main patch (e.g., audio files, images, text). At least the main patch should be completely authored by the student. If third-party material (e.g., software, images, audio) is used, then credit needs to be given (e.g., as a comment in the patch or in a separate text file) indicating authorship and source (as possible, e.g., website where it was downloaded from). If third-party material has been modified, the modifications have to be mentioned as comments in the patch.

B. The project has to run correctly in at least one of these: Pd Vanilla 0.48-0, Pd-Extended 0.42.5 (e.g., Juan's computer) or Pd-Extended 0.43.4 (e.g., computers in M210) for Mac OS.

C. Clear user instructions shall be provided within the patch (e.g., as comments).

D. ASSIGNMENT 6: Present a first version of the project to the class in the second session. The project has to be explained and problems encountered will be discussed. Opinions and suggestions will be given. This first version has to be submitted to the folder Ass_6 in Optima, before the second session.

E. ASSIGNMENT 7: After the second session a final version has to be uploaded to the folder Ass_7 in Optima , within 7 days.

GRADING:

Grading of assignments 6 and 7 will be done after 7 days from the second session, on a scale from 1 to 5 as explained below:

GRADE	ASSIGNMENT 6	ASSIGNMENT 7
1	Does not meet requirements, is untidy and no presentation is made or provided.	Does not meet requirements, is untidy and difficult to run.
2	Does not meet requirements, or it is untidy or no presentation is made or provided.	Does not meet requirements, or it is untidy or difficult run.
3	Meets requirements at least partially, is tidy and a presentation is made or provided.	Fully meets requirements (except D) .
Provided (3) is met, one extra point will be given for each of these:		
+1	Fully meets requirements A, B and D	Substantial improvement since ASSIGNMENT 6, advanced complexity or advanced efficient programming.
+1	Fully meets requirement C	The main patch opens as an easy-to-use Graphical User Interface with hidden connections.

Note that grading puts emphasis on getting the job done according to the requirements, tidiness and end-user experience.

Exceptionally, a student that is not able to attend the second session may send the project material (i.e., patch and associated files) and a video clearly explaining the patch and possible challenges. Additionally, a Skype session could be arranged. Support and tutoring will be available by e-mail (juigmend@student.jyu.fi) or at room M301. If needed, send your material for feedback before uploading it to Optima.

Pure Data Workshop
Advice

(not only useful for working with Pure Data)

1) Get the job done:
If you are asked to do *A*, you are expected to return *A*, not *B,* not *'It will work, but you need to also do this and this',* not *'I almost did it, but I need more time*', not *'I didn't get it done as requested... but I tried my best!'*. When a client or an employer is paying you for a product or service, that's exactly what you should return, not something else. Your client or employer will most likely fire you if instead of giving solutions you give more problems. Focus first in getting the job done. When you have secured that, go ahead and improve it, try alternatives or whatever else you want to do.

2) Re-use material:
If you have already done work or someone else has done work that you can use, then use it. It makes no sense to do existing things again from scratch to get a job done. It only makes sense in such cases as when you want to do it as an exercise, for pleasure or when because of copyright or something alike you cannot use someone else's work. Save time by re-using and hacking material. Also you can learn from other people's work.

3) Use documentation and other information resources:
To get a job done you don't need to know everything about a system, language or discipline. Memorizing things takes time and many repetitions. Apart from consulting other people's work, use documentation and help material, whether it is electronic or printed in paper. Ask Google. Most likely someone else already has had the same problem you are facing and most probably the solution is posted in a forum or other medium on the internet or in paper.

4) Break down the project into stages:
Make a plan for your project. Divide it in sections, milestones, modules or whatever you want to call them. Work on them sequentially or otherwise depending on which is better for you and for the project. At some point or points put things to work together. Start with the simpler tasks. This will make your work more efficient. Also it will be more enjoyable as on every stage you finish you will get a sense of accomplishment.

5) First the forest, then the trees, then the bushes:
Aim first to get a draft of the whole project, then go into details. Make drafts of the big sections of your project, put them together and see how they work together. Then improve them independently, put them together again and see again how the whole thing works. Repeat this process as needed. This will make the work efficient and to get to the desired results faster.

6) Don't be afraid to fail:
Consider failure as an opportunity to learn. If you fail then try again using a slightly different method or strategy. By doing this, each time you fail again you will *fail better*. If you continue failing and learning from your mistakes eventually you will converge to an optimal solution.