

21M.380 MUSIC AND TECHNOLOGY

SOUND DESIGN

Pd ASSIGNMENT 1 (PD1)

SEQUENCER

DUE: MONDAY, FEBRUARY 22, 2016, 9:30AM
SUBMIT TO: MIT LEARNING MODULES ▶ ASSIGNMENTS
5% OF TOTAL GRADE

1 Instructions

Build a music sequencer in Pure Data. If you do not know what a sequencer is, the *Online Sequencer*¹ will give you a practical understanding very quickly.

¹ <http://onlinesequencer.net/>

2 Restrictions

- Only Pd objects from Pd vanilla are allowed. Submissions that rely on objects from Pd extended will not be accepted!
- Messages, number boxes, GUI elements, symbols, arrays, tables, and comments of all kinds are allowed.
- Your patch should not rely on any external hardware besides keyboard and mouse. This rules out any MIDI controllers.

3 Guidelines

3.1 Programming guidelines

It might help to think of your sequencer as an engineering problem in three parts and address them in the following order:

Sound design Start by trying to build some interesting sounds that can be tested in isolation. Use the three simple sounds from figure 1 as a starting point and create additional sounds by adjusting the parameters in that patch as instructed. Try to introduce variety and come up with something that sounds interesting!

Time base Consult our main textbook² for inspiration on how to provide a time base that organizes the playback of your

² Farnell 2010.

HOW TO RUN THIS PATCH:

- 1 Turn on DSP by clicking this in run mode: `pd dsp 1`
- 2 In run mode, click either of the two [bang(messages or the square toggle box below
- 3 Adjust the numbers as instructed to create different sounds.

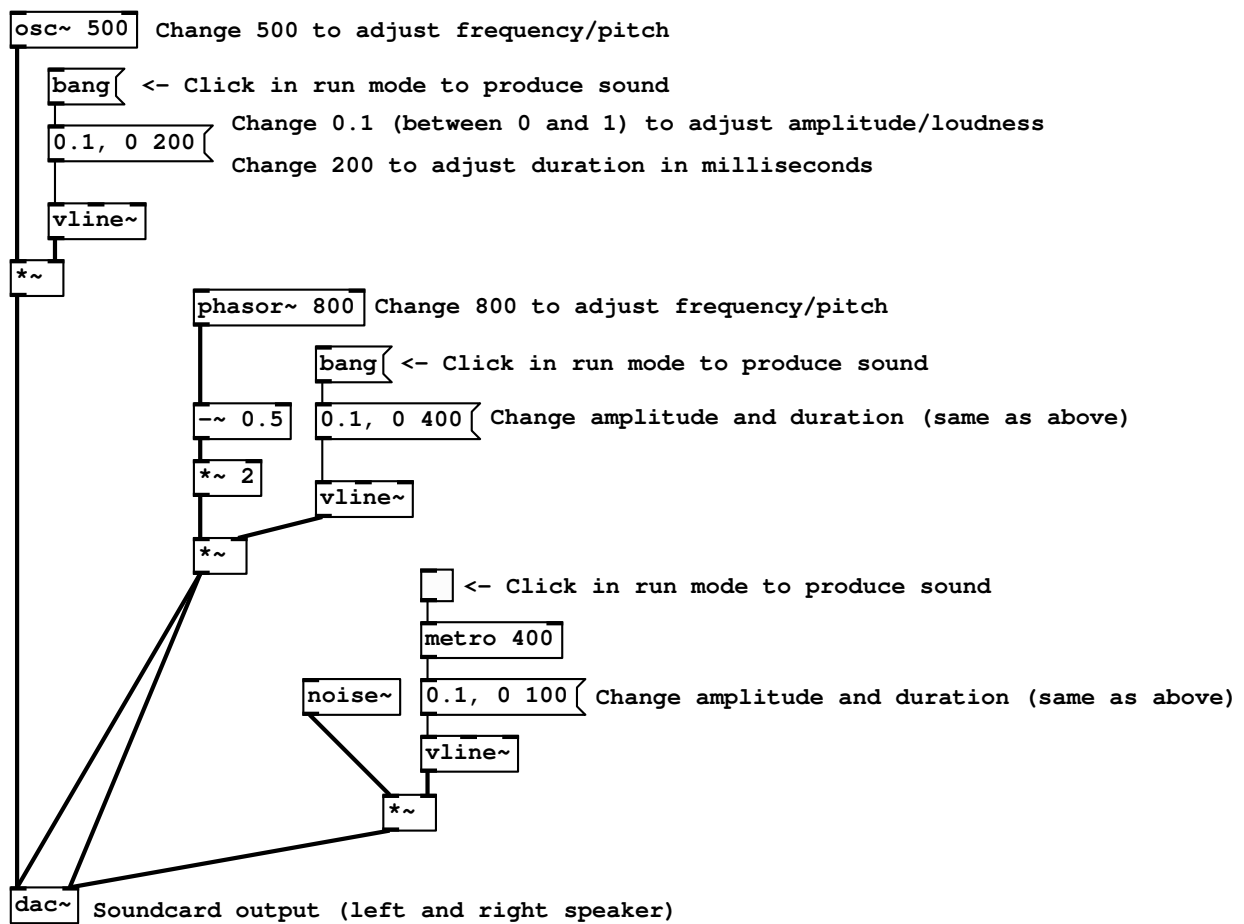



FIGURE 1. Some simple sounds in Pd

previously designed sounds along a timeline. The patch from figure 10.9 in the book provides a good basis along the lines of what we have discussed in class. What is the output of the [select] object in figure 10.9 of the book? How could you connect that output to the sounds you have designed in order to automate their playback? Section 14.3 introduces some advanced concepts that you can use to create a more complex time base for your assignment.

User interface What aspects of your sequencer should the user be able to control and how? Can one turn individual elements on and off? Toggle boxes () should be useful to design an interface that can be controlled with the mouse. If you feel adventurous, you can also check the help patch of the [key] object and use it together with [select] to build a sequencer that can be controlled with your computer keyboard. Resist the temptation to spend more time on GUI than on sound programming, though.

3.2 Musical guidelines

There is no need to restrict yourself to any particular musical genre. However, if you feel that you lack musical inspiration or experience, it might help to think of your sequencer as providing the following layers and then implement them in that order:

Rhythm in the form of a loop, comparable to the role of a drum kit in a band. Remember the patch that I built in the very first lecture for inspiration.

Harmony in form of less repetitive chords, comparable to the role of keyboards, synthesizers, and rhythm guitars.

Melody on top of everything else, comparable to the role of a singer or solo instrument.

4 Assessment criteria

Functionality Does your patch work flawlessly on another machine, potentially using a different OS, that runs Pd vanilla?

Readability Have you cleanly separated the user interface part of your patch from the rest of the code? Have you cleaned up the code before submission, removing experimental or debugging routines that are not actually required to run it?

Have you documented your intentions clearly by including descriptive comments (Put >> Comment)? If your patch requires initialization by the user (e.g., clicking [bang(messages, etc.)), have you clearly labeled which steps need to be completed in which order to get it to work? The Pd coding conventions³ provide useful suggestions for readability of your code.

³ <http://www.earcatching.com/pdconv/>

[Computer] programs must be written for people to read, and only incidentally for machines to execute.
(Abelson and Sussman 1996)

Creativity How interesting and diverse is the sound world that you have created?

Also, remember to forget everything you've been told and create something that you enjoy to program and listen to! ☺

5 Submission format

Submit your assignment as a single .pd file. Subpatches are allowed, abstractions are not.⁴

⁴ If you do not know what subpatches or abstractions are, don't worry, just make sure you submit a single .pd file.

References and useful resources

Abelson, Harold and Gerald Jay Sussman (1996). *Structure and interpretation of computer programs*. 2nd ed. MIT Press.
Farnell, Andy (2010). *Designing Sound*. Cambridge, MA and London: MIT Press. 688 pp. ISBN: 978-0-262-01441-0. MIT LIBRARY: [001782567](#). Hardcopy and electronic resource.

MIT OpenCourseWare
<http://ocw.mit.edu>

21M.380 Music and Technology: Sound Design
Spring 2016

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.