Chapter 6. Meeting 6, Foundations: Processing and Transforming Sounds

6.1. Announcements

- Bring amps and controllers to next class (tomorrow)
- Quiz next Monday
- Complete discussion leader assignment schedule is posted

6.2. Review Pd Tutorial 3

- ?

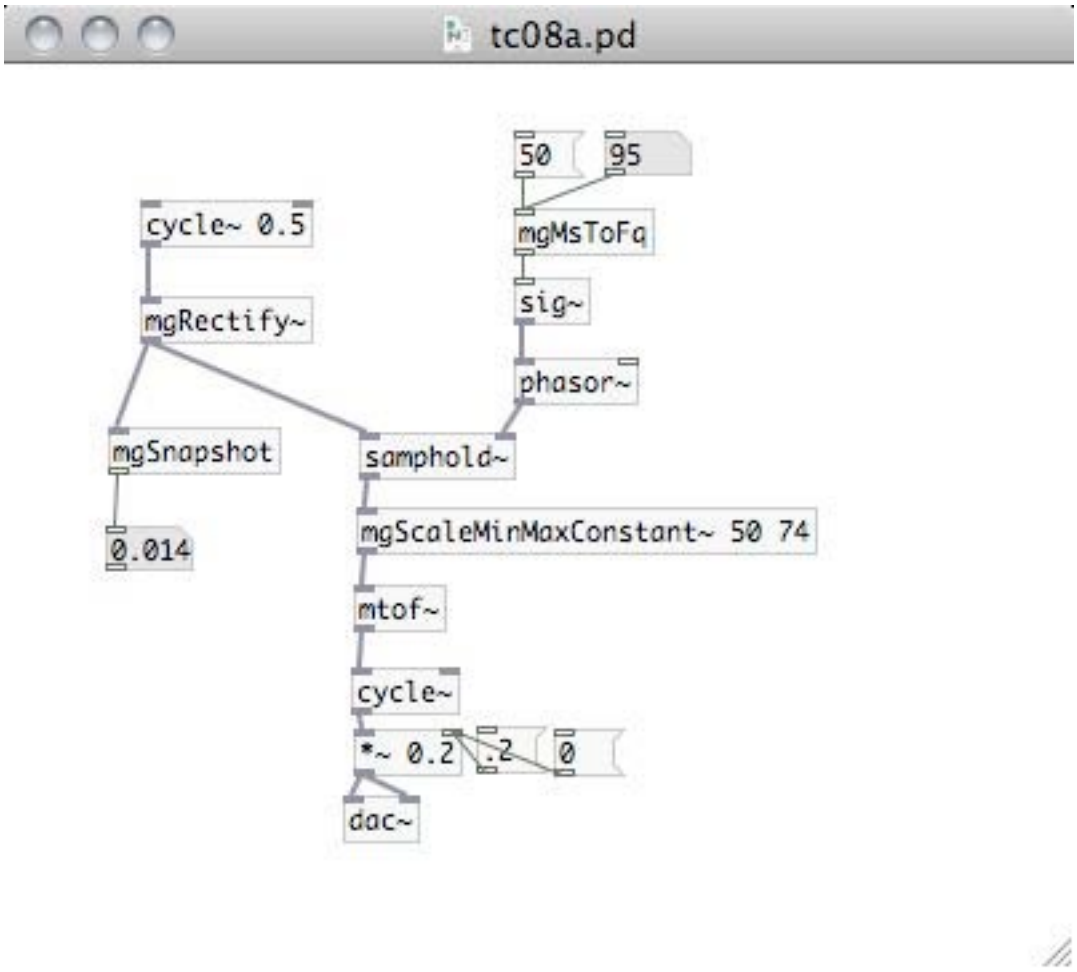
6.3. Reading: Wanderley and Orio: Evaluation of Input devices for Musical Expression: Borrowing Tools from HCI

- Wanderley, M. M. and N. Orio. 2002. "Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI." *Computer Music Journal* 26(3): pp. 62-76.
- What are some categories of input devices for musical expression the authors offer?
- What are some musical and aesthetic problems in evaluating musical controllers?
- What is Fritts's law? Does it relate to music making? Does it relate to the dual analog controller?
- How are musical controllers like controlling a vehicle?
- What are some contexts of interactive computer music mentioned?
- What ways can you think of quantitatively evaluating musical controllers?

6.4. Slowing Signals

- The audio rate is fast (44100 samples per second)
- [mgSnapshot]: periodically samples a signal and returns a value the event rate (once every 50 msec)
- Sometimes we want to sample a signal and return the value as a signal
- [samphold~]: sample a signal and output the value until triggered

- Trigger is when value in right inlet decreases (such as during one cycle of [phasor~])
- Using [samphold~] to map discrete values of a sine tone to the pitch of another sine tone.



6.5. Lists and Arrays

- A collection of similar data in a single row accessible by an index
- Lists count from 1

  Lists can be floats or symbols
- Arrays count from 0

  Arrays have to be floats

6.6. Arrays and Tables

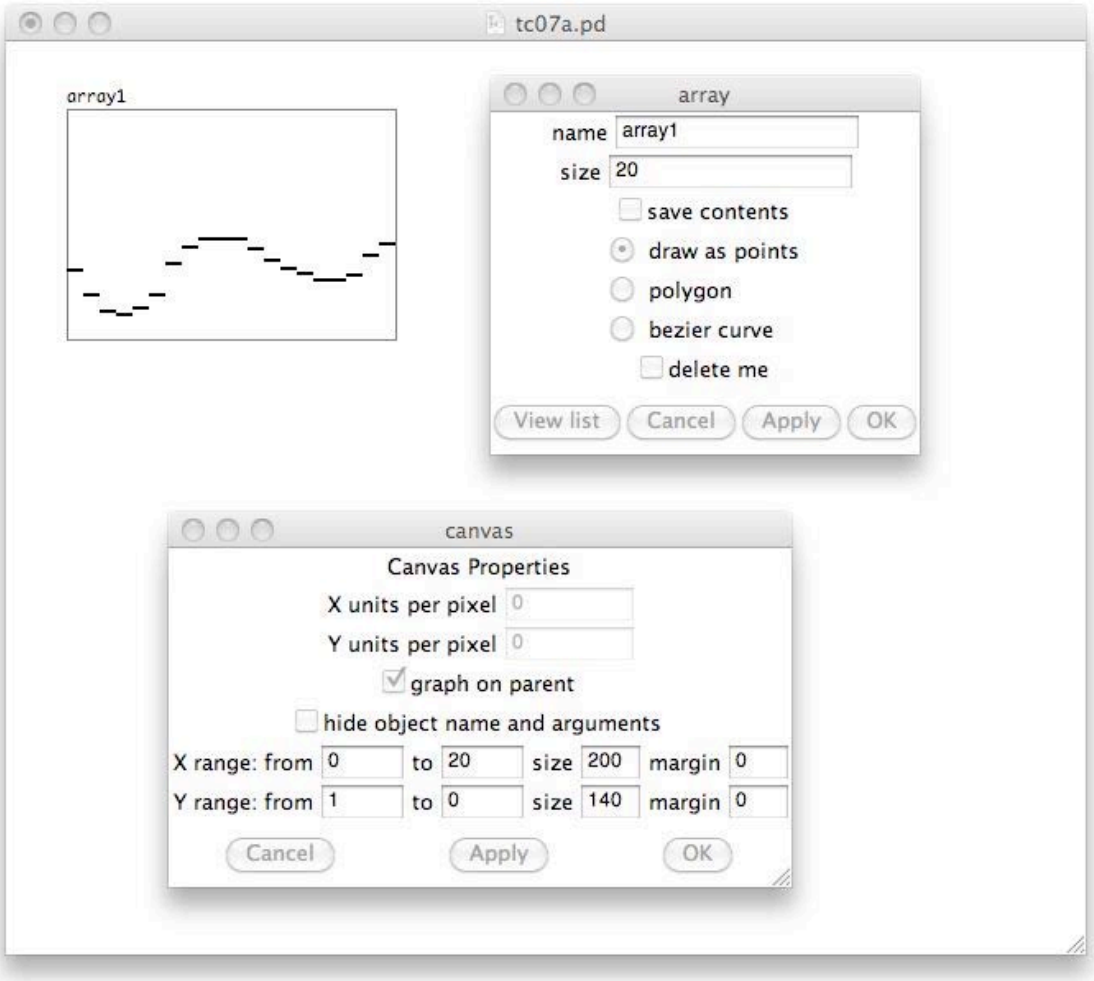
- Arrays store data in an object, and then access and write data into that object through other objects
- Data can be small collections of parameters, or millions of samples (an audio file)
- All arrays are named, though array names can be randomly generated with each patch using $0
- Array: visual representation of an array through a Canvas object
- Table: an Array Graph hidden in a sub-patch: can be given a dynamic name: [table name]

6.7. Arrays: Creating and Configuring

- Use the Put menu: Menu: Put > Array
- Will get windows for properties; can configure later by option-clicking
- Canvas parameters
  - Values here are for visual presentation only: does not limit the range of values that can be stored
  - Size is pixels on the screen
  - Range is mapping of values inside of the display
  - Y range is entered from max to minimum (not as expected)
- Array parameters
  - Name must be unique

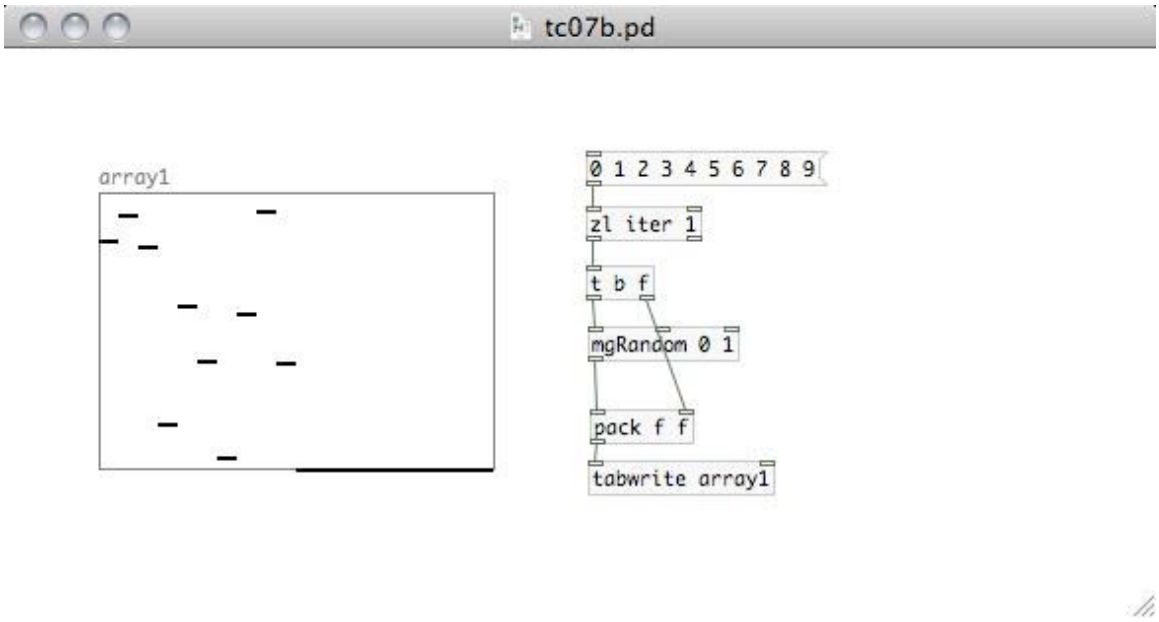
    Duplications result in an error: "warning: array1: multiply defined"

    Array names can be accessed from any open patch
  - Size: number of data points
  - Draw as points / polygon / bezier curve refers to presentation alone
  - Save content: stores data in patch (not reccommended: stores raw data in Pd patch)
- Use [table] for all arrays that do not need to be seen (most)
- Create an array with 20 points (array properties), and (under canvas properties) an X range from 0 to 20 and a Y range from 1 to 0 (note order of values)



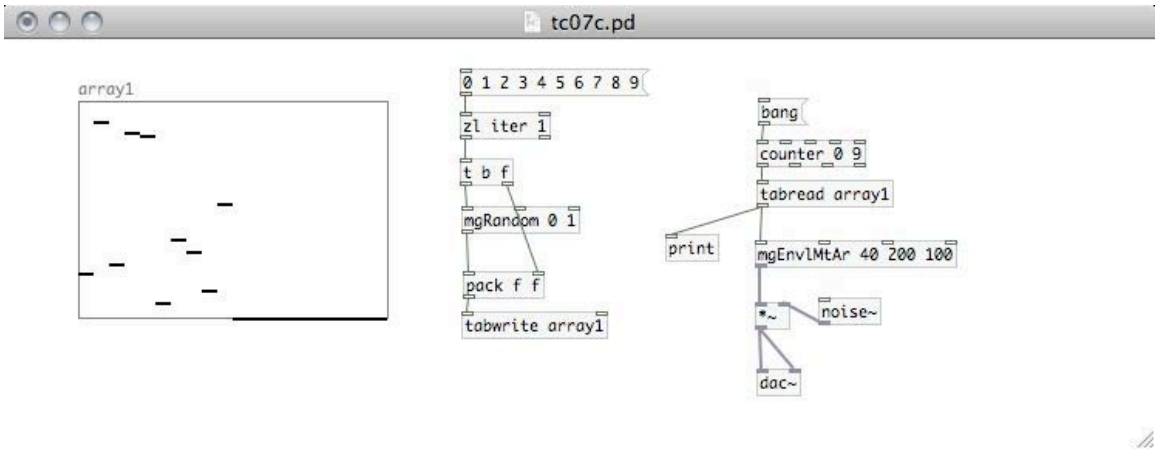
6.8. Adding Data

- Can directly edit graphical representation: click and drag in run mode
- [tabwrite name]: provide value, index pairs (as a list); must include array name as object argument
- [soundfiler]: given an array name, can load and resize an array for an audio file (we will use the more powerful [mgSoundfilerControl])
- Can dynamically build 2 element lists of value, index position:



6.9. Reading Array Data at the Event Rate

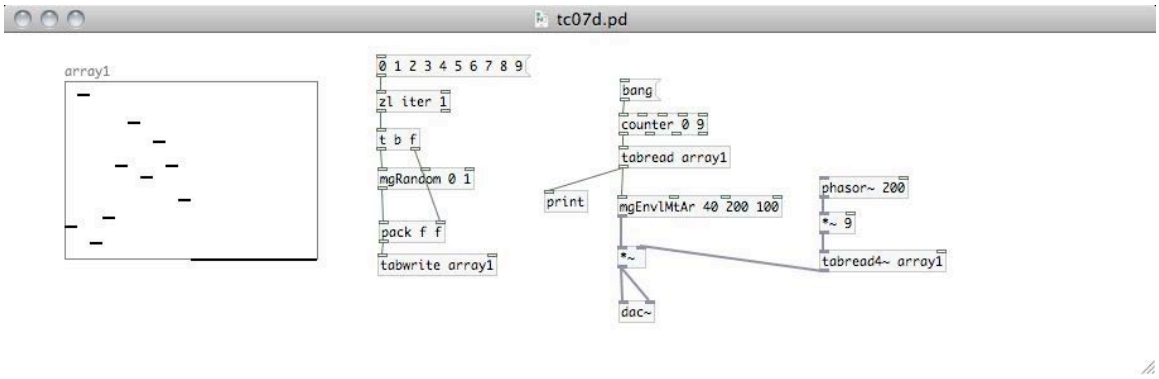
- [tabread name]: given an array name, provide index, get value
- Can read values from an array and employ them for peak envelope values and triggers



6.10. Reading Array Data at the Audio Rate

- Can read data from an array at the audio rate, using arrays to store wavetables or audio samples
- Input is a signal providing the index value, not a number; a scaled [phasor~] (from 0 to the size of the table-1) is the most common approach
- [tabread~ name]: read array data at the audio rate

- [tabread4~ name]: read array data at the audio rate with interpolation
- Using a small array as a wavetable with [tabread4~]



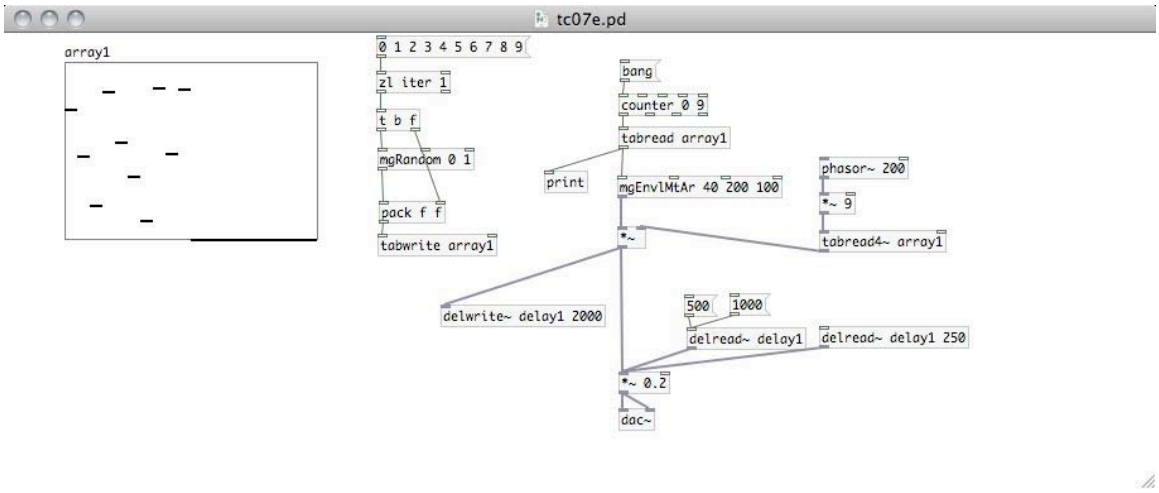
6.11. Delays and Reverbs

- Many common effects are created through combinations of delays
- Echos and reverbs are all built from delays
- Very short reflections blur together
- Echos interfere with the original signal, changing its timbre
- Example: martingale/demo/processorsDelay.pd

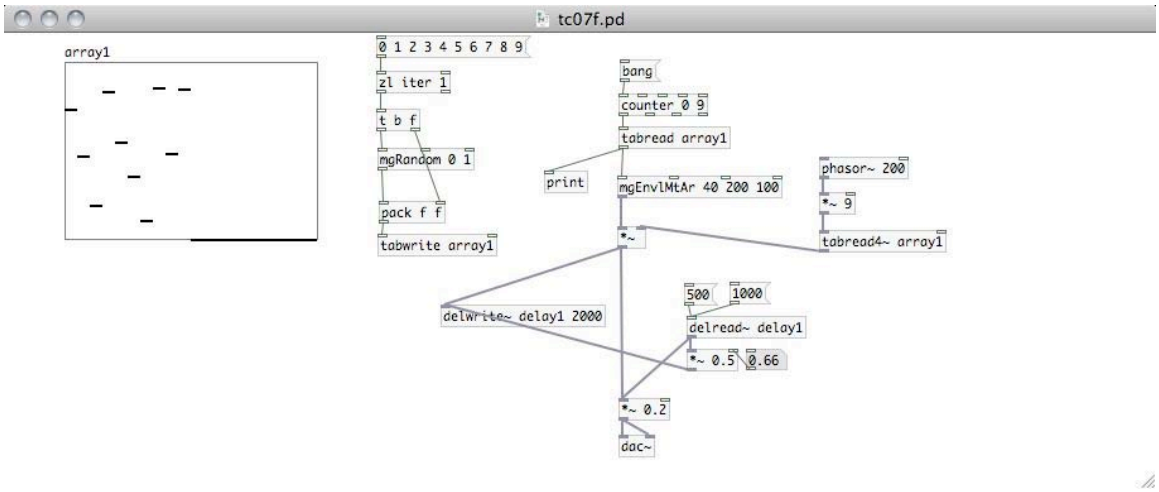
  Example: martingale/demo/processorsDelayDense.pd

6.12. Creating and Reading from Delay Lines

- Delay lines are stored memory for signals
- Must be declared with a name and a maximum delay time: [delwrite name 1000]
- Can be read from at numerous different times, and time can be provided as an argument: [delread~ name 200]



- Can create multiple echos by scaling signal and returning to delay line (feedback)



6.13. Listening: Musica Elettronica Viva (MEV)

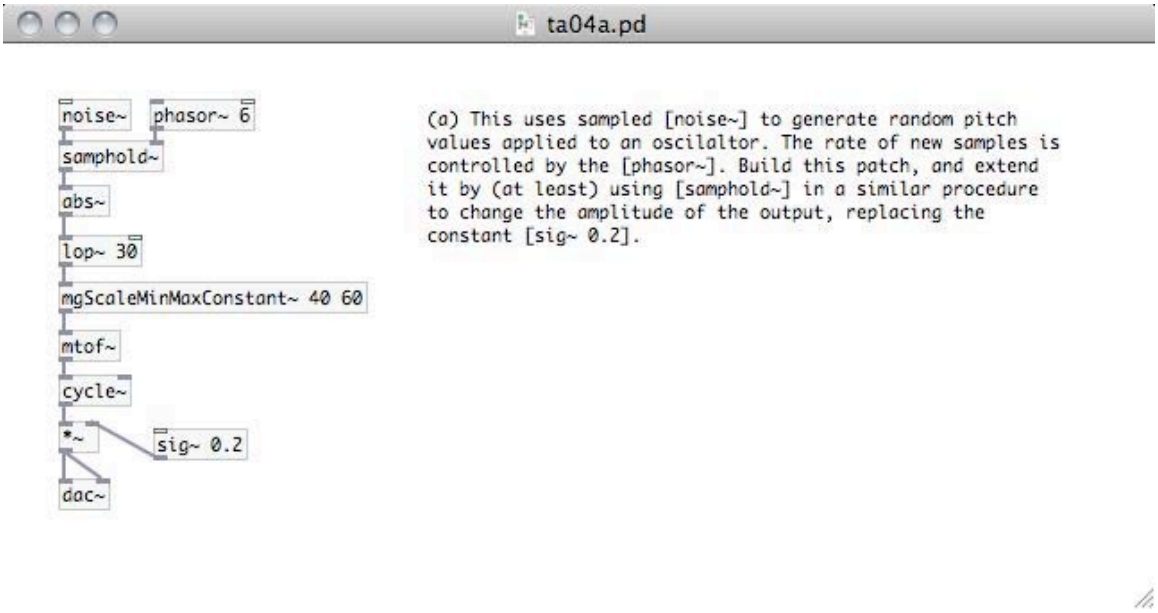
- Listening: Musica Elettronica Viva (MEV), *Spacecraft*, 1967

6.14. Listening: AMM

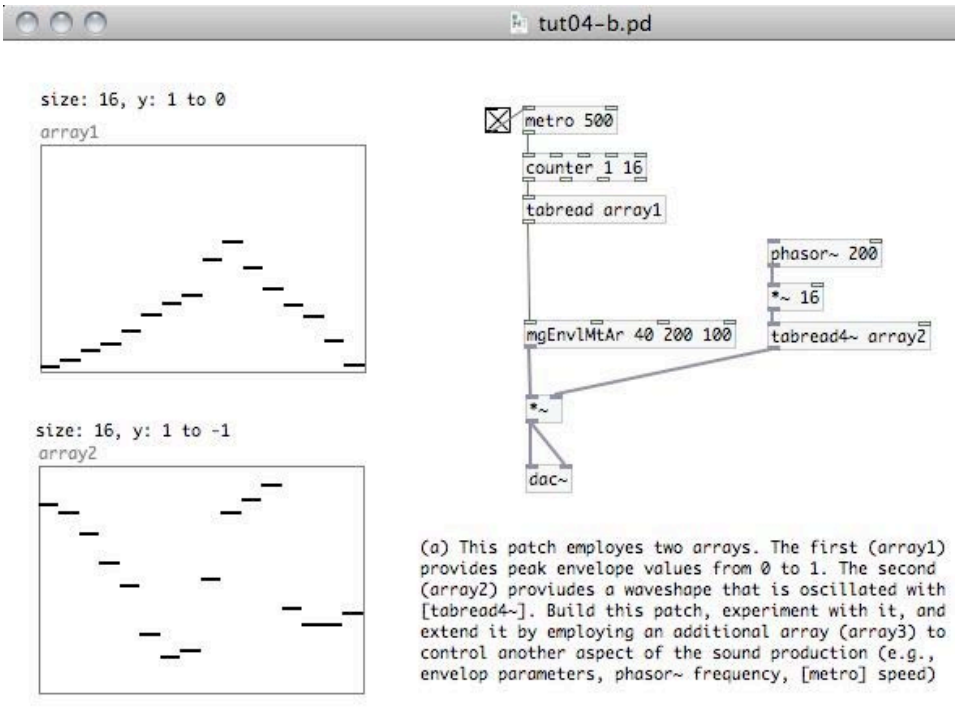
- Listening: AMM, "The Great Hall," *Laminal*, 1982

6.15. Pd Tutorial 4

1. Create and extend the following patch.



2. Create and extend the following patch.



(a) This patch employes two arrays. The first (array1) provides peak envelope values from 0 to 1. The second (array2) proviudes a waveshape that is oscillated with [tabread4~]. Build this patch, experiment with it, and extend it by employing an additional array (array3) to control another aspect of the sound production (e.g., envelop parameters, phasor~ frequency, [metro] speed)

21M.380 Music and Technology: Live Electronics Performance Practices
Spring 2011