Chapter 11. Meeting 11, Workshop

11.1. Announcements

- Musical Design Report 2 due today, 11 March
- Quiz next Tuesday
- Next week, and after spring break: make appointments with me to talk about sonic system projects.

11.2. Workshop: Musical Design Report 2

- `Three students presenting today`

11.3. Configuring Event Outputs

- For each EventMode, some output is always created (in EventMode midiPercussion, a midi file is always created)
- Some outputs are independent of any EventMode (an xmlAthenaObject can be always generated)
- A list of desired outputs (EventOutputs) is always consulted to determine what required and what optional outputs are created when creating an EventList
- The EOls command can be used to see what EventOutputs are active

```
:: eols
EventOutput active:
{name}
   acToolbox
   audioFile
   csoundBatch
   csoundData
   csoundOrchestra
   csoundScore
   midiFile
   pureDataArray
   superColliderTask
   textSpace
   textTab
   xmlAthenaObject
```

- When working with Csound, it is always desirable to have csoundData selected, as this causes the creation of integrated CSD files (combined orchestra and score files). Use the EOo to select active EventOutputs.

```
:: eoo cd
EventOutput formats: csoundData.

:: eols
EventOutput active:
{name}
   acToolbox
   audioFile
   csoundBatch
 + csoundData
   csoundOrchestra
   csoundScore
   midiFile
   pureDataArray
   superColliderTask
   textSpace
   textTab
   xmlAthenaObject
```

- It is desirable to write an AthenaObject XML file (xmlAthenaObject) when creating an EventList to premit reloading an athenaCL session.

```
:: eoo xao
EventOutput formats: csoundData, xmlAthenaObject.

:: eols
EventOutput active:
{name}
   acToolbox
   audioFile
   csoundBatch
 + csoundData
   csoundOrchestra
   csoundScore
   midiFile
   pureDataArray
   superColliderTask
   textSpace
   textTab
 + xmlAthenaObject
```

11.4. A Noise Instrument

- Csound instruments add auxiliary parameter fields to Textures
- Such parameter fields permit control of synthesis parameters
- Command sequence:
  - emo cn
  - tin a 13
  - tie r cs,(whps,e,(bg,rp,(5,10,15,20)),0,.200,.050)
  - *set initial low-pass filter cutoff frequency*

tie x2 whps,e,(bg,rp,(5,10,20,2,10)),0,400,18000

- *set final low-pass filter cutoff frequency*

  tie x3 whps,e,(bg,rp,(5,10,20,2,10)),0,400,18000

- *panning controlled by fractional noise with infrequent zero-order Markov controlled jumps out of 1/f2 to 1/f0*

  tie n n,100,(mv,a{2}b{0}:{a=12|b=1}),0,1

- eln; elh

11.5. A Sample Playback Instrument

- Csound instruments add auxiliary parameter fields to Textures
- Such parameter fields permit control of synthesis parameters
- Command sequence:
  - emo cn
  - tin a 32
  - *set a file path to an audio file*

    tie x6 cf,/Volumes/xdisc/_sync/_x/src/martingale/martingale/audio/29561.aif

  - *line segment absolute rhythm durations*

    tie r cs,(ls,e,(ru,5,30),(ru,.03,.15),(ru,.03,.15))

  - *start position within audio file in seconds*

    tie x5 ru,0,40

  - tie a ls,e,(bg,rc,(3,5,20)),.1,1
  - tie x2 whps,e,(bg,rp,(5,10,20,2,10)),0,100,10000
  - eln; elh

11.6. A Sample Playback Instrument with Variable Playback Rate

- Csound instruments add auxiliary parameter fields to Textures
- Such parameter fields permit control of synthesis parameters

- Command sequence:
  - emo cn
  - tin a 230
  - *set a file path to an audio file*

    tie x6 cf,/Volumes/xdisc/_sync/_x/src/martingale/martingale/audio/32673.aif
  - *line segment absolute rhythm durations*

    tie r cs,(ls,e,(ru,10,30),(ru,.05,.25),(ru,.05,.25))
  - *start position within audio file in seconds*

    tie x5 ru,0,10
  - *initial and final audio playback rate*

    tie x7 mv,a{1}b{.75}c{.5}d{.2}e{2}:{a=6|b=3|c=2|d=1|e=1}

    tie x8 mv,a{1}b{.75}c{.5}d{.2}e{2}:{a=6|b=3|c=2|d=1|e=1}
  - *panning controlled by fractional noise with infrequent zero-order Markov controlled jumps out of 1/f2 to 1/f0*

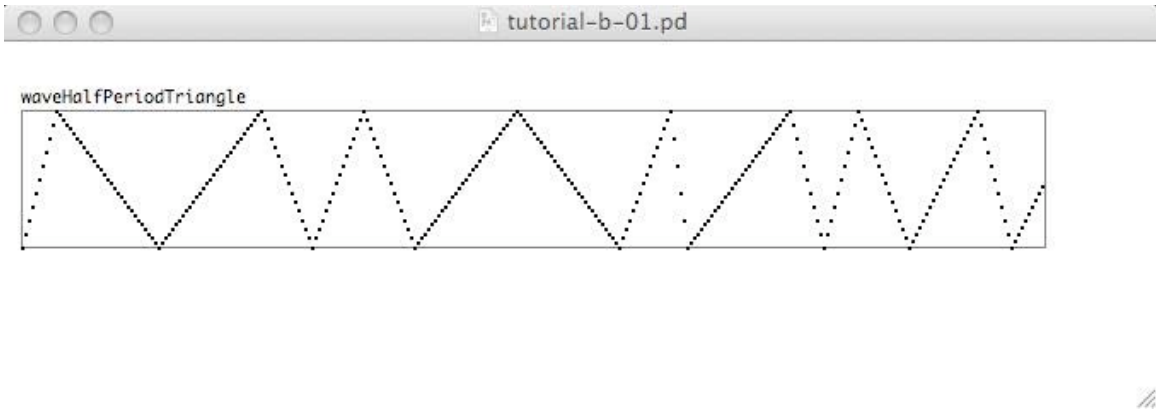
    tie n n,100,(mv,a{2}b{0}:{a=12|b=1}),0,1
  - *two instances simultaneously*

    ticp a b
  - eln; elh

11.7. Exporting ParameterObject output as PD Arrays

- ParameterObject outputs can be exported as PD Arrays.
- Can be used to produce control data used in PD processing
- The TPe (TextureParameter Export) command interactively

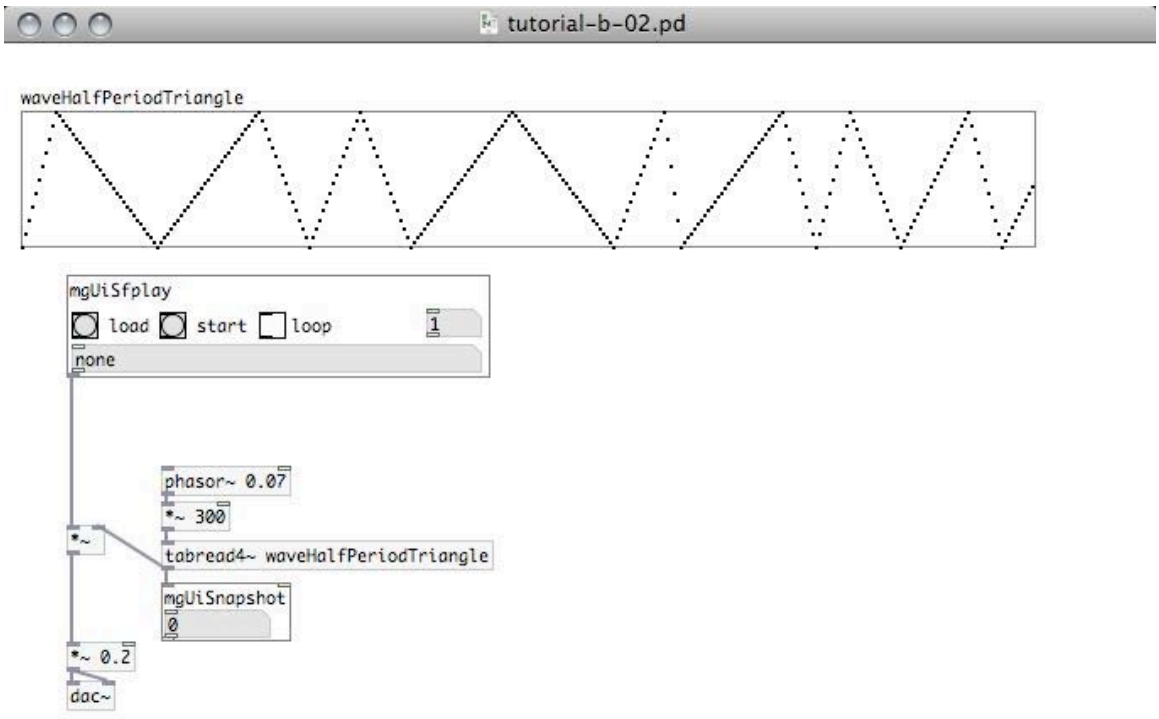
```
:: tpe
enter an export format: pda
number of events: 300
enter a Generator ParameterObject argument: whpt,e,(bg,rc,(5,10,15,20,30))
command.py: temporary file: /Volumes/xdisc/_scratch/ath2010.03.11.08.16.16.pd
complete: (/Volumes/xdisc/_scratch/ath2010.03.11.08.16.16.pd)
```



- Or as a single Command:
  - tpe pda 300 whpt,e,(bg,rc,(5,10,15,20,30))

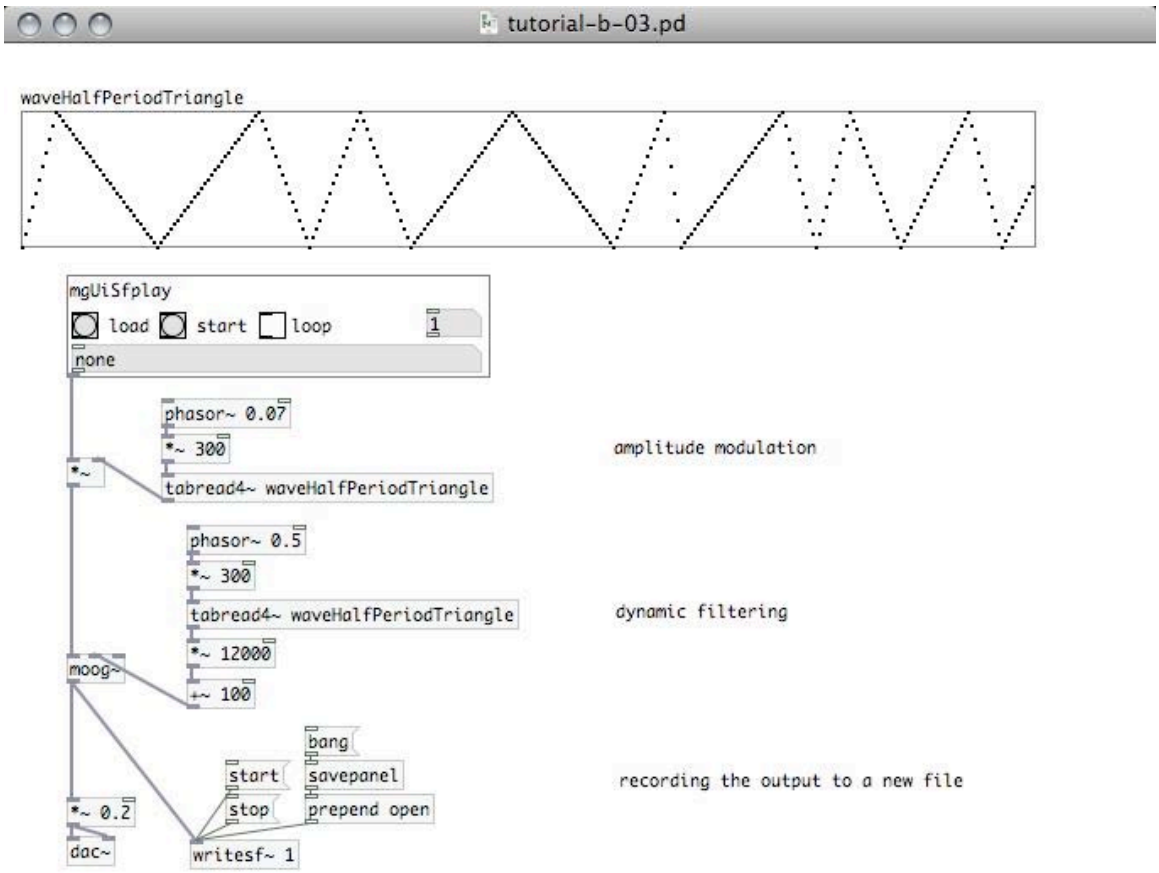
11.8. Using PD Arrays to Process Sound Files

- PD Arrays can be read at the audio rate by [tabread4~] objects
- The [tabread4~] object needs index values at the audio rate, best provided by a scaled [phasor~] (provide output cyclical output between 0 and 1)
- Scaling the amplitude by the PD Array



- Providing dynamic filtering with a [moog~] filter

  [writesf~] can be used to write a new audio file


tutorial-b-03.pd
waveHalfPeriodTriangle
mgUiSfplay
load
start
loop
1
none
phasor~ 0.07
*~ 300
*~
tabread4~ waveHalfPeriodTriangle
amplitude modulation
phasor~ 0.5
*~ 300
tabread4~ waveHalfPeriodTriangle
dynamic filtering
*~ 12000
moog~
+~ 100
bang
start
savepanel
recording the output to a new file
stop
prepend open
*~ 0.2
dac~
writesf~ 1

21M.380 Music and Technology: Algorithmic and Generative Music
Spring 2010