Chapter 8. Meeting 8, Approaches: Permutations, Generators, and Chaos

8.1. Announcements

- KIOKU concert this Friday, 6:30, in the MIT Lewis Music Library
- Musical Design Report 2 due 11 March: details to follow
- Sonic System Project Draft due 27 April: start thinking

8.2. A Line Segment Generator

- Often we need to vary a parameter linearly over time or events
- Break point functions require defining individual points
- LineSegment: A dynamic line generator between minimum and maximum values over a duration

```plaintext
:: tpmap 100 ls,e,30,0,1
lineSegment, (constant, 30), (constant, 0), (constant, 1)
TPmap display complete.
```

```plaintext
:: tpmap 100 ls,e,(bg,oc,(3,6,9)),(ru,0,.7),(ru,.3,1)
lineSegment, (basketGen, orderedCyclic, (3,6,9)), (randomUniform, (constant, 0), (constant, 0.7)), (randomUniform, (constant, 0.3), (constant, 1))
TPmap display complete.
```
8.3. Large Scale Amplitude Behavior with Operators

- By multiplying or summing multiple behaviors, dynamic large-scale shapes are possible
- Multiplying amplitudes by zero can create periods of inactivity
- Techniques derived from modular synthesis
- OperatorMultiply used to scale LineSegment and WavePulse

:: tpmap 120 om,(ls,e,9,(ru,.2,1),(ru,.2,1)),(wp,e,23,0,0,1)
operatorMultiply, (lineSegment, (constant, 9), (randomUniform, (constant, 0.2),
(constant, 1))), (randomUniform, (constant, 0.2), (constant, 1))),
(wavePulse, event, (constant, 23), 0, (constant, 0), (constant, 1))

TPmap display complete.

- Command sequence:
  - emo mp
  - tin a 64
  - tie r pt,(bg,rp,(16,16,8)),(bg,rp,(2,2,1,4)),(c,1)
  - tie a om,(ls,e,9,(ru,.2,1),(ru,.2,1)),(wp,e,23,0,0,1)
  - eln; elh

- Related ParameterObjects: OperatorAdd, OperatorMultiply, OperatorDivide, OperatorPower,
  OperatorSubtract, OperatorCongruence

8.4. Reading: Ames. A Catalog of Sequence Generators: Accounting for Proximity, Pattern, Exclusion, Balance and/or Randomness

- What does ames mean by dependence, exclusion, and balance
• Why does Ames have so many varieties of random uniform generators, such as LEHMER, SPREAD, FILL, and others?

• How is Brownian motion related to random walks?

• How does Ames characterize the artistic opportunities of using 1/f noise?

• What are the characteristics of output provided by chaotic generators such as LOGISTIC and BAKER

• What is the idea of a chaos knob?

8.5. Continuous Random Walks

• We can use BasketGen for discrete random walks

• We can use Accumulator for continuous random walks

• Accumulator

```plaintext
:: tpmap 100 a,.5,(ru,-.1,.1)
accumulator, 0.5, (randomUniform, (constant, -0.1), (constant, 0.1))
TPmap display complete.
```

8.6. Chaos and the Logistic Map

• Complex dynamical systems

• Deterministic systems that exhibit complex behavior

• Most employ iterative processing and result in sensitivity to initial conditions (butterfly effect)

• The logistic map was developed as a model of population growth by Pierre Verhulst

\[ x_{n+1} = rx_n (1 - x_n) \]

\( r \) is a positive number between 0 and 4 that represents a combined rate for reproduction and starvation
- States produces constant outputs, oscillating behavior, and complex behavior

- LogisticMap: most interesting output available from \( p \) (or \( r \), lambda, or chaos knob) between 2.75 and 4

```hlql
.: tpmap 100 lm,.5,(ls,e,100,2.75,4),0,1
logisticMap, 0.5, (lineSegment, (constant, 100), (constant, 2.75), (constant, 4)), (constant, 0), (constant, 1)
TPmap display complete.
```

- Related ParameterObjects: henonBasket, lorenzBasket
8.7. Reading: Voss and Clarke. 1/f Noise in Music: Music from 1/f Noise


- What is a 1/f noise, and what is the variations of noise from 1/f0, 1/f1, 1/f2, 1/f3?

- The sound and shape of correlated noise: [noiseColors.pd]

- What technique did Voss and Clarke use to analyze music?

- What sort of data did Voss and Clarke collect?

- Extracting an amplitude envelope from an audio signal: [vossClarke.pd]

- What conclusions do Voss and Clarke make about 1/f spectral densities?

- Is music (or the averaged spectral analysis of amplitude envelopes) intelligent behavior

- What technique did Voss and Clarke use to generate melodies? Is this technique parallel to the analysis technique?

- What conclusions did they draw from human evaluation of their generated melodies? Were these conclusions based on the 1/f noise source?

8.8. 1/f Noise

- Rather than just one type of 1/f noise, use many

- Gamma can move between 0 (white), 1 (pink), 2 (brown), 3 (black)

- 1/f noise: gamma == 1

```
:: tpmap 100 n,100,1,0,1
noise, 100, (constant, 1), (constant, 0), (constant, 1)
TPmap display complete.
```

- 1/f noise: gamma == 2
8.9. 1/f Noise in Melodic Generation: LineGroove

- Using BasketSelect to select discrete values from a continuous generator

```
:: tpmap 100 bs,(2,4,7,9,11,14,16,19,21,23),(n,100,1,0,1)
basketSelect, (2,4,7,9,11,14,16,19,21,23), (noise, 100, (constant, 1),
```
(constant, 0), (constant, 1)),
TPmap display complete.

- Command sequence using TM LineGroove:
  - emo m
  - tmo lg
  - tin a 108
  - tie r cs,[(ls,e,10),(ru,.01,.2),(ru,.01,.2)]
  - tie f bs,(2,4,7,9,11,14,16,19,21,23),(n,100,1,0,1)
  - cln; clh

8.10. 1/f Noise in Melodic Generation: HarmonicAssembly

- Command sequence using TM Harmonic Assembly:
  - emo m
  - pin a d3,e3,g3,a3,b3,d4,e4,g4,a4,b4,d5,e5,g5,a5,b5
  - tmo ha
  - tin a 27
  - tie r pt,(c,16),(lg,(bg,rc,(1,2,3,5,7)),(bg,rc,(3,6,9,12))),(c,1)
  - tie a om,[(ls,e,9),(ru,2,1),(ru,2,1)],(wp,e,23,0,0,1)
  - tie d0 c,0
  - tie d1 n,100,2,0,14
  - tie d2 c,1
  - tie d3 c,1
8.11. Tutorial: PD Arrays as Parameters: Filtered Noise

- \([\text{mgEnvMtAr}]\) creates a mono-triggered, attack-release envelope

\([\text{mgEnvMtAr}]\) arguments: attack time, release time, duration

\([\text{mgEnvMtAr}]\) trigger: a floating point value that sets the peak amp

8.12. Tutorial: PD Arrays as Parameters: Cyclical Amplitude Values

- Looping through an array with amplitude values with \([\text{counter}]\)

\([\text{pow 4}]\) provides non-linear to linear amplitude scaling
8.13. Tutorial: PD Arrays as Parameters: Cyclical Cutoff Frequency Values

- Looping through an array if values scaled to MIDI pitch values (60-140) with [mgScaleMap]
  - MIDI pitch values are scaled to frequency values with [mtof]
  - Data values are converted to signals with [mtof] and [lop~ 30]
  - [moog~] provides a signal controlled low pass filter with variable resonance

- [metro] provides regularly spaced triggers

[counter] and [sel 1] are used to select the first of each cycle

An array of values is scaled to pulse multipliers with [mgScaleMap]

The [counter] max value is dynamically set after reading and mapping a value from the array
21M.380 Music and Technology: Algorithmic and Generative Music
Spring 2010

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