



ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΣΧΟΛΗ ΜΟΥΣΙΚΗΣ & ΟΠΤΙΚΟΑΚΟΥΣΤΙΚΩΝ ΤΕΧΝΩΝ
ΤΜΗΜΑ ΕΘΝΟΜΟΥΣΙΚΟΛΟΓΙΑΣ

Εργασία 1: Ψηφιακή Επεξεργασία Ήχου

Ναταλία Γ. Λιθαδιώτη – 2448

Διδάσκων: Βασίλης Αγιομυριανάκης

ΛΗΞΟΥΡΙ, Σεπτέμβριος 21

Κατασκευάστε 20 ήχους με χρήση τεχνικών όπως προσθετική σύνθεση και Phase Vocoder στο SuperCollider. Για την προσθετική σύνθεση μπορείτε να χρησιμοποιήσετε περιοδικούς ταλαντωτές όπως για παράδειγμα, ημιτονοειδής, πριονωτή, τριγωνική κυματομορφή κλπ, καθώς και γεννήτριες θορύβου (noise generators) όπως λευκός θόρυβος, ροζ θόρυβος κλπ. Χρησιμοποιήστε ηχογραφημένους ήχους για Phase vocoder.

Οι ήχοι αυτοί θα χωρίζονται σε 2 χρονικές κατηγορίες:

- sustained (10 sounds)
- short (10 sounds)

Πειραματιστείτε με την περιβάλλουσα (envelope) του ήχου για να χτίσετε το σχήμα και τη διάρκεια του.

[TIP] Ψάξτε στο help του supercollider και στο ίντερνετ για παραδείγματα κώδικα και πειραματιστείτε.

Όλοι οι ήχοι θα είναι σε SynthDef και θα έχουν περιβάλλουσα (envelope) και comments με επεξήγηση του κώδικα.

Η άσκηση θα είναι σε .pdf μαζί με το .scd αρχείο και θα πρέπει να έχει στην πρώτη σελίδα το Όνομα, Α.Μ, τίτλο κλπ, σύμφωνα με τον κανονισμό της Σχολής Μουσικής και Οπτικοακουστικών Τεχνών του Ιονίου Πανεπιστημίου. Στην δεύτερη σελίδα θα υπάρχει και η εκφώνηση της άσκησης.

*Θα ανεβάσετε τον κώδικα στο φάκελο DSP στο GitHub σε φάκελο με το όνομα Ergasia1.

*Δείτε τις παρακάτω οδηγίες σχετικά με τις εργασίες στο GitHub.

s.boot;

//----- 10 Short sounds-----//

```

//1

s.boot;

(
  SynthDef(\name1,{|freq = 220, gate = 1, amp = 0.5|
    var source, env, sound1;
    source = SinOsc.ar(freq, 0, amp);
    env = EnvGen.kr(Env.perc,doneAction:2);
    sound1 = source * env;
    Out.ar([0,1], sound1)
  }).play;
)

//2

s.boot;

(
  SynthDef(\name2, {|freq = 400, gate = 1, amp = 0.9|
    var source, env, sound2;
    source = (SinOsc.ar(400, 0, 0.9)+
      SinOsc.ar(880, 6, 0.02));
    env = EnvGen.kr(Env.perc(0.1, 0.8, 1, -4),doneAction:2);
    sound2 = source * env;
    Out.ar([0,1], sound2 * BrownNoise.ar(0.1))
  }).play;
)

//3

s.boot;

```

```
(  
  SynthDef(\name3, {|freq = 800, gate = 1, amp = 0.6|  
    var source, env, sound3;  
    source = (SinOsc.ar(400, 0, 0.9)+  
      Saw.ar(880, 0.02));  
    env = EnvGen.kr(Env.perc(1, 0.1, 1, -4),doneAction:2);  
    sound3 = source * env;  
    Out.ar([0,1], sound3 * WhiteNoise.ar(0.25))  
  }).play;  
)
```

//4

s.boot;

```
(  
  SynthDef(\name4, {|freq = 200, gate = 1, amp = 0.7|  
    var source, env, sound4;  
    env = EnvGen.kr(Env.perc(0.1, 0.3, 0.5, -3.14),doneAction:2);  
    source = (SinOsc.ar(880, 0, 0.02)+  
      SinOsc.ar(440, 0, 0.5)+  
      Saw.ar(880, 0.15));  
    sound4 = source * env;  
    Out.ar([0,1], sound4)  
  }).play;  
)
```

//5

s.boot;

```
(
  SynthDef(\name5, {|freq = 200, gate = 1, amp = 0.7|
    var source, env, sound5;
    env = EnvGen.kr(Env.perc(1, 0.5, 1, -3),doneAction:2);
    source = (SinOsc.ar(880, 0, 0.02)+
      Saw.ar(440, 0.05)+
      Saw.ar(880, 0.15));
    sound5 = source * env;
    Out.ar([0,1], sound5 * PinkNoise.ar(3))
  }).play;
)
```

//6

s.boot;

```
(
  SynthDef(\name6, {|amp = 0.4|
    var s6, env, p, ssum;
    s6 = Saw.ar(LFNoise0.kr(2).exprange (100, 200), amp);
    p = Pulse.ar(ExpRand(20, 400));
    env = Line.kr(0, 1, 5, doneAction:2);
    ssum = (s6 * p) + env;
    Out.ar([0,1], ssum)
  }).play;
)
```

//7

s.boot;

```
(  
SynthDef(\sound7, {|freq = 250, amp = 0.20|  
    var s7, env, ssum;  
    s7 = SinOsc.ar(freq, 0, amp);  
    env = Line.kr(0, 0.20, 4, doneAction:2);  
    ssum = s7 / env;  
    Out.ar([0,1], ssum)  
}).play;  
)
```

//8

s.boot;

```
(  
SynthDef(\sound8, {|freq = 230, amp = 0.4|  
    var s8, env, ssum;  
    s8 = SinOsc.ar(freq, 0, amp);  
    env = Line.kr(0.75, 1.5, 4, doneAction:2);  
    ssum = s8 + env;  
    Out.ar([0,1], ssum)  
}).play;  
)
```

//9

s.boot;

```
(  
SynthDef(\sound9, {|freq = 60, amp = 0.3|  
    var s9, env, ssum;
```

```

        s9 = Saw.ar(freq, amp);

        env = Line.kr(0, 1, 4, doneAction:2);

        ssum = (s9 + env) + WhiteNoise.ar(0.25);

        Out.ar([0,1], ssum)

    }).play;

)

//10

s.boot;

(

    SynthDef(\sound10, {

|freq1 = 500, freq2 = 220, dur1 = 0.1, dur2 = 0.02, cur1 = 1, cur2 = (-1), atk
    = 0.01, rel = 1, c1 = 1, c2 = (-12), amp = 0.5|

        var s10, env, ssum;

        s10 = Env([freq1, freq2], [dur1, dur2],[cur1, cur2]).ar;

        env = Env([0, 1, 0],[atk, rel], [c1, c2]).kr(2);

        ssum = SinOsc.ar(s10, pi/2);

        ssum = ssum * env;

        ssum = Pan2.ar(ssum, 0, amp);

        Out.ar([0,1], ssum);

    }).play;

)

//-----10 sustained sounds-----//

//1

```

```

s.boot;

(
  SynthDef(\n1,{|freq = 220, gate = 1, amp = 0.5|
    var source, env, s1;
    source = SinOsc.ar(880, 0, 0.2);
    env = EnvGen.kr(Env.adsr(1, 0.5, 1, 1, 2, -3, 0),doneAction:2);
    s1 = source * env;
    Out.ar([0,1], s1
  }).play;
)

```

//2

```

s.boot;

(
  SynthDef(\n2, {|freq = 400, gate = 1, amp = 0.9|
    var source, env, s2;
    source = (SinOsc.ar(400, 0, 0.9)+
    SinOsc.ar(880, 6, 0.02));
    env = EnvGen.kr(Env.adsr(0.1, 0.8, 1, -4),doneAction:2);
    s2 = source * env;
    Out.ar([0,1], s2 * BrownNoise.ar(0.1))
  }).play;
)

```

//3

```

s.boot;

(

```



```

SynthDef(\n3, {|freq = 800, gate = 1, amp = 0.6|
    var source, env, s3;
    source = (SinOsc.ar(400, 0, 0.9)+
        Saw.ar(880, 0.02));
    env = EnvGen.kr(Env.linen(1, 0.1, 1, -4),doneAction:2);
    s3 = source * env;
    Out.ar([0,1], s3 * WhiteNoise.ar(0.25))
}).play;
)

```

//4

s.boot;

(

```

SynthDef(\n4, {|freq = 200, gate = 1, amp = 0.7|

```

```

    var source, env, s4;

```

```

env = EnvGen.kr(Env.linen(0.1, 0.3, 0.5, -3.14),doneAction:2);

```

```

source = (SinOsc.ar(880, 0, 0.02)+

```

```

    SinOsc.ar(440, 0, 0.5)+

```

```

    Saw.ar(880, 0.15));

```

```

s4 = source * env;

```

```

    Out.ar([0,1], s4)

```

```

}).play;

```

)

//5

s.boot;

(

```

SynthDef(\n5, {|freq = 200, gate = 1, amp = 0.7|
    var source, env, s5;
env = EnvGen.kr(Env.adsr(1, 0.5, 1, -3),doneAction:2);
    source = (SinOsc.ar(880, 0, 0.02)+
        Saw.ar(440, 0.05)+
        Saw.ar(880, 0.15));
    s5 = source * env;
    Out.ar([0,1], s5 * PinkNoise.ar(3))
    }).play;
    )

//6
s.boot;
(
    SynthDef(\phaseV, {
        var in, chain, chain2, env, sound20;
        in = BrownNoise.ar(0.8);
env = EnvGen.kr(Env.perc, doneAction:2);//Line.kr(0.75, 1.5, 4);
        chain = FFT(LocalBuf(2048), in);
chain2 = PV_RectComb(chain, 0.95, LFTri.kr(0.097, 0, 0.4, 0.5) +
        LFTri.kr(0.24, 0, -0.5, 0.5));
        sound20 = (in*env + chain2);
        Out.ar(0, IFFT(sound20))
    }).add;
    )
Synth(\phaseV);

```

```

//7

s.boot;

(
  SynthDef(\sound7, {|freq = 90, amp = 0.7|
    var s6, s7;

    s6 = SinOsc.ar(freq, 0, amp);
s7 = SinOsc.ar(freq, 0.7, amp) * Saw.ar(freq/2, amp/2);

    Out.ar(0, s6);
    Out.ar(1, s7);
  }).play;
)

```

```

//8

s.boot;

(
  SynthDef(\sound8, {
    var s8, s9, freq1, freq2, amp1, amp2;
    freq1 = LFNoise0.kr(4).exprange(150, 300);
    freq2 = LFNoise0.kr(4).exprange(150, 300);
    amp1 = LFPulse.kr(4, 0, 0.10) * 0.55;
    amp2 = LFPulse.kr(4, 0.5, 0.10) * 0.55;
    s8 = LFPulse.ar(freq1, 0.5, amp1);
    s9 = LFPulse.ar(freq2, 0.5, amp2);
    s8 = FreeVerb.ar(s21, 0.7, 0.8, 0.25);
    s9 = FreeVerb.ar(s22, 0.7, 0.8, 0.25);

    Out.ar(0, s8);
    Out.ar(1, s9);
  }
)

```

```

    }).play;

    )

//9

s.boot;

(

  SynthDef(\sound9, {

    var s10, s11, freq1, freq2, amp1, amp2;
    freq1 = LFNoise0.kr(4).exprange(200, 350);
    freq2 = LFNoise0.kr(4).exprange(200, 350);

    amp1 = Saw.kr(4, 0, 0.12) * 0.75;
    amp2 = Saw.kr(4, 0.5, 0.12) * 0.75;
    s10 = LFPulse.ar(freq1, 0.5, amp1);
    s11 = LFPulse.ar(freq2, 0.5, amp2);
    s10 = FreeVerb.ar(s31, 1, 0.8, 0.25);
    s11 = FreeVerb.ar(s32, 1, 0.8, 0.25);

    Out.ar(0, s10);
    Out.ar(1, s11);

  }).play;

  )

//10

s.boot;

(

  SynthDef(\soun10, {

    var s12, s13, freq1, freq2, amp1, amp2;
    freq1 = LFNoise0.kr(4).exprange(100, 380);

```

```
freq2 = LFNoise0.kr(4).exprange(100, 380);  
    amp1 = SinOsc.kr(4, 0, 0.12) * 0.75;  
    amp2 = SinOsc.kr(4, 0.5, 0.12) * 0.75;  
    s12 = LFPulse.ar(freq1, 0.5, amp1);  
    s13 = LFPulse.ar(freq2, 0.5, amp2);  
    s12 = FreeVerb.ar(s41, 0.85, 0.8, 0.25);  
    s13 = FreeVerb.ar(s42, 0.85, 0.8, 0.25);  
    Out.ar(0, s12);  
    Out.ar(1, s13);  
}).play;  
)
```