

リズム

聴覚の時間分解能

音響旋律のエクササイズ「3秒の旋律」で、「3秒」という時間が、人間の聴覚における「経験のひとかたまり(チャンク)」であり、「聴覚的現在」の長さである、と述べた。それでは、人間の耳は、音の流れを時間方向にどのくらい細かく分けて聴くことができるのだろうか。人間の耳の時間的な解像度のことを、聴覚の時間分解能と呼ぶ。人間の耳の時間分解能は、おおよそ以下のように階層化されている。

同時 : 約1000分の3秒(約3[ms]、約300[Hz])、おおよそ15サンプル以下の間隔で起こる事象は聴覚系では「同時」と認識される。

```
// 300[Hz]の点列
~out = { Impulse.ar(300.dup, 0, 0.5, 0) };
```

```
// TaskProxyの定義
x = TaskProxy.basicNew;
~out = \points;
x.play;
```

// 0~0.005秒の任意の間隔で左右の耳に別の点を鳴らす。2点の鳴る時間間隔と「同時」の認識の関連を確かめてみよ。

```
(
var d, p;
x.source = {
  loop {
    p = [-1, 1].choose.postln;
    d = rrand(0.0, 0.005).postln;
    ~out.spawn([\pan, p, \sustain, 0.1]);
    d.wait;
    ~out.spawn([\pan, -1 * p, \sustain, 0.1]);
    3.wait;
  }
}
)
```

順序 : 2音が約100分の3秒(約30[ms]、約30[Hz]=可聴周波数の下限)、おおよそ150サンプル以上離れると、その順序を聴きとることができる。

```
// 30[Hz]の点列
~out = { Impulse.ar(30.dup, 0, 0.5, 0) };
```

// 0.005~0.05秒の任意の間隔で左右の耳に別の点を鳴らす。2点の鳴る時間間隔と「順序」の認識の関連を確かめてみよ。

```
~out = \points;
(
var d, p;
x.source = {
  loop {
    p = [-1, 1].choose.postln;
    d = rrand(0.005, 0.05).postln;
    ~out.spawn([\pan, p, \sustain, 0.1]);
    d.wait;
    ~out.spawn([\pan, -1 * p, \sustain, 0.1]);
    3.wait;
  }
}
)
```

反応 : 2音が約10分の3秒(約300[ms])以上離れると、最初の音刺激に対して「反応」することができる。

```
// 3[Hz]の点列
~out = { Impulse.ar(3.dup, 0, 0.5, 0) };
```

// 0.05~0.5秒の任意の間隔で左右の耳に別の点を鳴らす。

```
~out = \points;
(
```

```

var d, p;
x.source = {
  loop {
    p = [-1, 1].choose.postln;
    d = rrand(0.05, 0.5).postln;
    ~out.spawn([\pan, p, \sustain, 0.1]);
    d.wait;
    ~out.spawn([\pan, -1 * p, \sustain, 0.1]);
    3.wait;
  }
}
)

```

(注)人間の意識は脳が出したイリュージョンであり、知覚してから意識が生まれるまでに、約0.2秒の時間を必要とする。

次の音はいくつに聞こえるか？

```

// 0~0.05秒の任意の間隔で左右の耳に0~10個の点を鳴らす
(
var d, p;
x.source = {
  loop {
    (1 + 10.rand).postln.do {
      p = 1.rand2.postln;
      d = rrand(0.001, 0.2).postln;
      ~out.spawn([\pan, p, \sustain, 0.1]);
      d.wait;
    };
    3.wait;
  }
}
)

```

音の順序を思い出すことができるか？

点だけでなく、線や面も混ぜてみる。

```

// 点/線/面のアサイン
~points = \points;
~line = \line;
~plane = \plane;
~out = ~points + ~line + ~plane;

// 0~0.05秒の任意の間隔/持続時間で左右の耳に0~10個の点/線/面を鳴らす
(
var d, p;
x.source = {
  loop {
    (1 + 10.rand).postln.do {
      d = 0.05.rand;
      switch ( 3.rand.postln,
        0, { ~points.spawn([\amp, 0.9, \freq, 1, \pan, 1.rand2, \sustain, d]) },
        1, { ~line.spawn([\amp, 0.1, \freq, [1000, 796].choose, \pan, 1.rand2, \sustain, d]) },
        2, { ~plane.spawn([\amp, 0.1, \pan, 1.rand2, \sustain, d]) }
      );
      d = (d + 0.05.rand).postln;
      d.wait;
    };
    3.wait;
  }
}
)

x.stop;

```

旋律の伸縮

▼エクササイズ：伸長/圧縮された旋律

まず最初にエンヴェロープを使って構成された3秒間の音響旋律をひとつ用意せよ。次にその旋律のエンヴェロープを時間的に伸縮させ、どのように聴こえ方が変化するかを確かめよ。

```
// 振幅最小(ゼロ)から最大まで1.5秒で増加、そののちに同じく1.5秒で最小まで減少する面による旋律
(
  var aarray = [0, 1, 0]; // 振幅の変化(三角形のエンヴェロープ)
  var tarray = [1.5, 1.5]; // 増加減少時間
  ~out = {
    var enva = Env.new(aarray, tarray);
    Pan2.ar(
      WhiteNoise.ar(EnvGen.kr(enva, doneAction:2)), 0.0, 0.5
    )
  }
)

// 増加減少の時間を0~10秒の間でさまざまに変化させる
(
  var aarray = [0, 1, 0]; // 振幅の変化(三角形のエンヴェロープ)
  var tarray = [[0.1, 0.2, 0.5, 1, 3, 5, 8, 10].choose, [0.1, 0.2, 0.5, 1, 3, 5, 8, 10].choose].postln; // 増加減少時間
  ~out = {
    var enva = Env.new(aarray, tarray);
    Pan2.ar(
      WhiteNoise.ar(EnvGen.kr(enva, doneAction:2)), 0.0, 0.5
    )
  }
)
```

同じことをインタラクティブに行なう。

```
// エンヴェロープのトリガー用プロキシ
~trig.kr(1);
~trig = { Impulse.kr(0.5) }; // トリガー間隔=2秒
(
  var aarray = [0, 1, 0]; // 振幅の変化(三角形のエンヴェロープ)
  var tarray = [1.0, 1.0]; // 増加減少時間
  ~out = {
    var enva = Env.new(aarray, tarray);
    Pan2.ar(
      WhiteNoise.ar(EnvGen.kr(enva, ~trig.kr, timeScale:MouseX.kr(0.001, 1.0, 1))), 0.0, 0.5
    )
  }
)
```

時間を伸縮させることによって、印象が次々と変化していくような(あるいはあまり変化しないような)旋律をデザインせよ。

```
// 振幅とパンが100回ランダムに変化する面による0~6秒の旋律(100セグメントのエンヴェロープを使用)
(
  var aarray = Array.fill(101, {0.5.sum3rand + 0.5}); // 振幅はガウス分布で変動
  var parray = Array.fill(101, {1.0.sum3rand}); // パンも同じくガウス分布で変動
  var t = 0.06.rand.postln; // 時間間隔はランダムに決定(0~0.06秒)
  ~out = {
    var enva = Env.new(aarray, Array.fill(100, t));
    var envp = Env.new(parray, Array.fill(100, t));
    Pan2.ar(
      WhiteNoise.ar(0.5), EnvGen.kr(envp, doneAction:2), EnvGen.kr(enva, doneAction:2)
    )
  }
)

// インタラクティブ・リスニング
~trig.kr(1);
~trig = { Impulse.kr(0.2) }; // トリガー間隔=5秒
(
  var aarray = Array.fill(100, {0.5.sum3rand + 0.5}) ++ [0]; // 振幅はガウス分布で変動
  var parray = Array.fill(101, {1.0.sum3rand}); // パンも同じくガウス分布で変動
  ~out = {
    var enva = Env.new(aarray, Array.fill(100, 0.03));
    var envp = Env.new(parray, Array.fill(100, 0.03));
  }
)
```

```

    var ts = MouseX.kr(0.1, 1.0, 1);
    Pan2.ar(
      WhiteNoise.ar(0.5), EnvGen.kr(envp, ~trig.kr, timeScale:ts), EnvGen.kr(enva, ~trig.kr,
timeScale:ts)
    )
  }
)

// ランダムに周波数が上昇していく線による0~30秒の旋律(100セグメントのエンヴェロープを使用)
(
var farray = Array.series(101, 4000, 10) + Array.rand(101, -20.0, 20.0); // 上昇する等差数列にランダムな変動を
付加
var parray = Array.fill(101, {1.0.sum3rand}); // パンはガウス分布で変動
var t = 0.3.rand.postln; // 時間間隔はランダムに決定(0~0.3秒)
~out = {
  var envf = Env.new(farray, Array.series(100, t, 0));
  var envp = Env.new(parray, Array.series(100, t, 0));
  Pan2.ar(
    SinOsc.ar(EnvGen.kr(envf, doneAction:2)), EnvGen.kr(envp, doneAction:2), 0.5
  )
}
)

// インタラクティブ・リスニング
~trig.kr(1);
~trig = { Impulse.kr(0.2) }; // トリガー間隔=5秒
(
var farray = Array.series(101, 4000, 10) + Array.rand(101, -100.0, 100.0); // 上昇する等差数列にランダムな変動
を付加
var parray = Array.fill(101, {1.0.sum3rand}); // パンはガウス分布で変動
var aarray = Array.fill(100, {0.2.sum3rand + 0.3}) ++ [0]; // 振幅もガウス分布で変動
~out = {
  var envf = Env.new(farray, Array.fill(100, 0.05));
  var envp = Env.new(parray, Array.fill(100, 0.05));
  var enva = Env.new(aarray, Array.fill(100, 0.05));
  var ts = MouseX.kr(0.001, 1.0, 1);
  Pan2.ar(
    SinOsc.ar(EnvGen.kr(envf, ~trig.kr, timeScale:ts)), EnvGen.kr(envp, ~trig.kr, timeScale:ts),
EnvGen.kr(enva, ~trig.kr, timeScale:ts)
  )
}
)
)

```

反復

2つの音が連続して鳴るだけで、そこにリズムを聴きとることができる。

```

// 0.5秒間隔で鳴る2つの点
~out = ~points;
~points.set(\freq, 2, \sustain, 1);
~points = \points;

```

2つの点の時間間隔を少しずつ長くしていく。時間間隔と(身体の動きを連想させるような)リズム感の間には、どのような関係があるか?

```

// 0.01~5秒の間隔で鳴る2つの点
~points.set(\amp, 0.9, \freq, 1, \pan, 0, \sustain, 0.1);
x.play;
(
var d = 0.01;
x.source = {
  loop {
    d.postln;
    ~points.spawn;
    d.wait;
    ~points.spawn;
    d = d * 1.5;
    if (d > 5, {d = 0.01});
    5.wait;
  }
}
)

```

```

// 点列の間隔を連続的に変化させる(点列の周波数のスweepと同様)
(
var d = 0.01;

```

```

x.source = {
  loop {
    d.postln;
    ~points.spawn;
    d.wait;
    ~points.spawn;
    d = d * 1.5; // 点列の間隔を3/2倍ずつにしていく(比による構成)
    if (d > 3, {d = 0.01});
    d.wait;
  }
}
)

```

ランダムな時間間隔で2つの点を鳴らす。

```

// 0.01~5秒のランダムな間隔で鳴る2つの点
(
var d = exprand(0.01, 5);
x.source = {
  loop {
    d.postln;
    ~points.spawn;
    d.wait;
    ~points.spawn;
    d = exprand(0.01, 5);
    5.wait;
  }
}
)

```

点列(旋律)の群化

たとえ、均等な時間間隔のパターンの場合でも、耳は数個の音をグルーピングして聴く傾向がある。このことを「主観的リズム化」という。2、3、4(=2+2)、6(=3+3、2+2+2)、8(=2+2+2+2)個の音からなる群が安定して知覚されることが多い。

```

// 20~40個の点からなる均等な時間間隔(0.01~5秒)のパターン
(
var d = exprand(0.01, 1);
x.source = {
  loop {
    d.postln;
    (20 + 20.rand).postln.do {
      ~points.spawn;
      d.wait;
    };
    d = exprand(0.01, 1);
    5.wait;
  }
}
)

x.stop;

```

反復するパターン

たとえランダムなパターンであったとしても、それを反復して聴くことでパターンを発見(広義のグルーピング)することができる。パターンの開始点(必ずしも鳴り始めとは限らない)は、どのように認識されるのだろうか?

```

// 8個の点からなるランダムなパターンの反復
(
var d = Array.exprand(8, 0.01, 1).postln;
~out = Pbind(
  \instrument, \points,
  \amp, 0.9,
  \freq, 1,
  \pan, Pfunc({1.rand2}),
)
)

```

```

        \sustain, 0.1,
        \dur, Pseq(d, inf)
    );
)

```

点の数を4倍(32個)にするとどのようにパターンを認識できるか?

```

// 32個の点からなるランダムなパターンの反復
(
var d = Array.exprand(32, 0.01, 1).postln;
~out = Pbind(
    \instrument, \points,
    \amp, 0.9,
    \freq, 1,
    \pan, Pfunc({1.rand2}),
    \sustain, 0.1,
    \dur, Pseq(d, inf)
);
)

```

音の数が多くなりすぎると、反復によるパターンの認識がしづらくなる。

高速リズムとしての低周波

120[bps]のリズムは2[Hz]の音、20[Hz](可聴域下限)の音は1200[bps]のリズムである。リズムから音への遷移領域のパターンを聴いてみる。音の時間解像度とリズムから音への遷移の関係はどうなっているか?

```

// 点の間隔がしだいに短くなっていく(点列の周波数がしだいに高くなっていく)パターン
(
var f = Array.geom(16, 1, 1.3).postln;
~out = Pbind(
    \instrument, \points,
    \amp, 0.9,
    \freq, Pseq(f, inf),
    \pan, 0,
    \sustain, 1,
    \dur, 1
);
)

```

モジュール：2と3によるリズム構成

人間は前述の主観的リズム化によって、均質なパルスを2ないし3をベースとしたまとまりに群化して知覚する傾向がある。そこで(フォボナッチ数列もそうであったように)、この2と3という基本要素を組み合わせていくことによるリズム構成を行う。

ここで「モジュール」という考え方を導入する。モジュールは「構造を構成する基準となる部品あるいは独立したユニット」のことである。ここでは2と3によるリズム・パターンをひとつのモジュールとして、それをボトムアップに構築していくことで、複雑なパターンを生成することを試みる。

```

// 規則的な4つのパルス(4拍子)上での2と3による構成
(
var p = Pseq([ 1, 1, 0, 0 ]); // 2によるモジュール
var q = Pseq([ 1, 1, 1, 0 ]); // 3によるモジュール
~out = Pbind(
    \instrument, \line,
    \amp, Prand([ p, q ], inf) * 0.5, // 2つのモジュールのランダムに実行
    \freq, 8820,
    \pan, 0,
    \sustain, 0.01,
    \dur, 0.2
);
)

// 2拍子と3拍子の2つのモジュールのランダムな実行
(
var p = Pseq([ 1, 0.2 ]); // 2によるモジュール
var q = Pseq([ 1, 0.2, 0.2 ]); // 3によるモジュール
~out = Pbind(
    \instrument, \line,

```

```

        \amp, Prand([ p, q ], inf) * 0.5,
        \freq, 8820,
        \pan, 0,
        \sustain, 0.01,
        \dur, 0.2
    );
)

// 2拍子と3拍子の2つのモジュールのランダムな実行
(
var p = Pseq([ 1.5, 1.5 ]); // 2によるモジュール
var q = Pseq([ 1, 1, 1 ]); // 3によるモジュール
~out = Pbind(
    \instrument, \line,
    \amp, 0.5,
    \freq, 8820,
    \pan, 0,
    \sustain, 0.01,
    \dur, Prand([ p, q ], inf) * 0.1
);
)

// 2拍子と3拍子の2つのモジュールをさらに2あるいは3回ずつ反復する
(
var p = Pseq([ 1, 0.2 ], { [2, 3].choose }); // 2によるモジュール
var q = Pseq([ 1, 0.2, 0.2 ], { [2, 3].choose }); // 3によるモジュール
~out = Pbind(
    \instrument, \line,
    \amp, Prand([ p, q ], inf) * 0.5,
    \freq, 8820,
    \pan, 0,
    \sustain, 0.01,
    \dur, 0.2
);
)

```

#キング・クリムゾンの「太陽と戦慄 PartII」のようなリズムになる。

```

// 1拍子のモジュールを加え1~3回ずつ反復する
(
var p = Pseq([ 1 ], { rrand(1, 3) }); // 1によるモジュール
var q = Pseq([ 1, 0.2 ], { rrand(1, 3) }); // 2によるモジュール
var r = Pseq([ 1, 0.2, 0.2 ], { rrand(1, 3) }); // 3によるモジュール
~out = Pbind(
    \instrument, \line,
    \amp, Prand([ p, q, r ], inf) * 0.5,
    \freq, 8820,
    \pan, 0,
    \sustain, 0.01,
    \dur, 0.2
);
)

// 1拍子のモジュールだけ反復数を増やす
(
var p = Pseq([ 1 ], { rrand(1, 8) }); // 1によるモジュール
var q = Pseq([ 1, 0.2 ], { rrand(1, 3) }); // 2によるモジュール
var r = Pseq([ 1, 0.2, 0.2 ], { rrand(1, 3) }); // 3によるモジュール
~out = Pbind(
    \instrument, \line,
    \amp, Prand([ p, q, r ], inf) * 0.5,
    \freq, 8820,
    \pan, 0,
    \sustain, 0.01,
    \dur, 0.2
);
)

// 2と3によるモジュールを振幅、周波数、時間のすべてのパラメータに適用
(
var p1 = Pseq([ 1, 0.2 ]); // 2によるモジュール(振幅)
var q1 = Pseq([ 1, 0.2, 0.2 ]); // 3によるモジュール(振幅)
var p2 = Pseq([ 1, 0.5 ]); // 2によるモジュール(周波数)
var q2 = Pseq([ 1, 0.5, 0.5 ]); // 3によるモジュール(周波数)
var p3 = Pseq([ 1.5, 1.5 ]); // 2によるモジュール(時間)
var q3 = Pseq([ 1, 1, 1 ]); // 3によるモジュール(時間)
~out = Pbind(

```

```

    \instrument, \line,
    \amp, Prand([ p1, q1 ], inf) * 0.5,
    \freq, Prand([ p2, q2 ], inf) * 441,
    \pan, 0,
    \sustain, 0.01,
    \dur, Prand([ p3, q3 ], inf) * 0.1
  );
)

```

サイレンス(沈黙)

「サイレンスは第4の音響素材である」

サイレンス(沈黙)の発見-それは音楽におけるゼロの発見にも等しい。たとえ物理的な振動としての音波がなくとも、たとえ周囲にさまざまな環境音がなくとも、サイレンスには音響的な持続が必ず存在している。それはある音の状態A(サイレンスの開始)から状態B(サイレンスの終了)へと向う音響旋律でもある。すべての音は旋律であるとすれば、すべてのサイレンスも旋律である。

通常の音の認知において、音のある部分(振幅パラータが0でない部分)が図(figure)であり、それ以外の部分は地(ground)であるとされる。サイレンスを意識する、ということは音響における図と地を反転するということでもある。2つの音が0.03秒間隔で鳴らされる場合、その2つの音の間には、0.03秒の持続を持ったサイレンスという旋律が生まれている。

```

// 10秒間隔で鳴る孤立した点、あるいは10秒の沈黙旋律(孤立した点は沈黙に挟まれている)
~out = { Impulse.ar(0.1.dup, 0, 0.8, 0) };

```

```

// 毎秒1回、0.01秒の無音部を含む面(面を中断する沈黙)
~out = { WhiteNoise.ar((EnvGen.ar(Env.linen(0, 0.99, 0, 1), Impulse.kr(1), 0.2, 0))).dup };

```

```

// 毎秒1回、0.99秒の無音部を含む面(沈黙を中断する面)
~out = { WhiteNoise.ar((EnvGen.ar(Env.linen(0, 0.01, 0, 1), Impulse.kr(1), 0.2, 0))).dup };

```

```

// 1秒の面の中におけるサイレンスの時間をマウスのX座標にマッピング(図と地の関係に注意せよ)
~out = { WhiteNoise.ar((EnvGen.ar(Env.linen(0, MouseX.kr(0.0, 1.0, 0), 0, 1), Impulse.kr(1), 0.2, 0))).dup
};

```

デジタル・オーディオの場合、アナログ・レコードやカセットテープの音を特徴づけているサーフェスノイズやヒスノイズはない。その高いS/N比によって、(人間の聴覚の能力との比較において)大幅に無音に近い状況を作り出すことが可能になった。

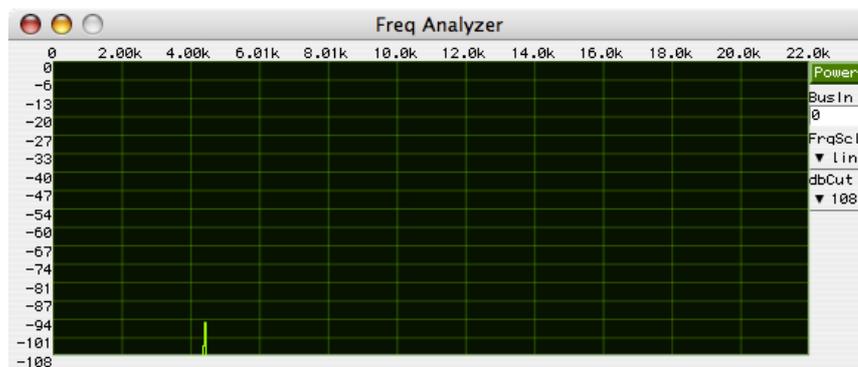
(注) とはいえ、CDのダイナミックレンジは約6万5千倍(96[dB])であり、人間の耳が持っている約100万倍(120[dB])ものダイナミックレンジには遠く及ばない。

無音が限りなく無音に近づくことで、有音と無音の差だけでなく、微音という極めて非常に微かな音や、存在してはいるが知覚できない(しづらい)可聴域を超えた音も、音響作品の構成要素のひとつとなり得る。沈黙には、物理的に音波が存在しない「数値的サイレンス」と聴覚的に音波が知覚できない「知覚的サイレンス」、そしてたとえそこに音があったとしてもあたかも無音のように聴こえる「認知的サイレンス」の3つがある。

```

// 最小の音(-96[dB]、441[Hz]の線)
~out = { SinOsc.ar(4410.dup, 0, 1/65536, 0)};

```



```

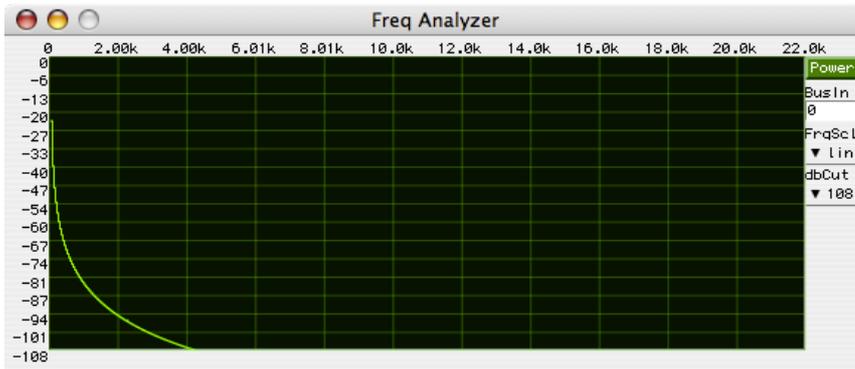
// 可聴域外の大容量

```

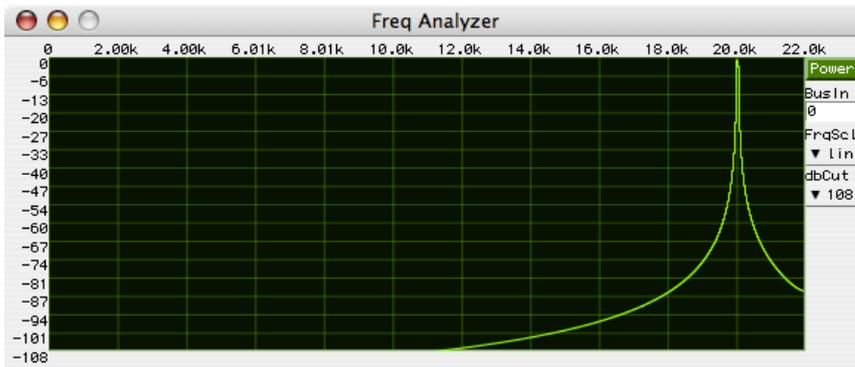
```

~out = { SinOsc.ar(15.dup, 0, 1, 0)}; // 振幅1、15[Hz]の線

```



```
~out = { SinOsc.ar(20000.dup, 0, 1, 0)}; // 振幅1、20000[Hz]の線
```



▼エクササイズ：ロウワーケース・サウンド(小文字の音)

微小な振幅の音と可聴域外の音のみを用いたコンポジションを行ってみよ。

▼エクササイズ：5つの点と1つの面によるコンポジション

5つの点と1つの面を使ったコンポジションを行ってみよ。

```
// TaskProxyの定義と点・線・面のアサイン
x = TaskProxy.basicNew;
~points = \points;
~line = \line;
~plane = \plane;
~out = ~points + ~line + ~plane;
x.play;
```

非常に短いサイレンスで結ばれた点と面の関係を調べてみる。

```
// 10秒の面の直後(0.05~0.5秒後)に1つの点を鳴らす
(
var d;
x.source = {
loop {
~plane.spawn([\amp, 0.1, \pan, 0, \sustain, 10]);
d = 10 + rrand(0.05, 0.5).postln;
d.wait;
~points.spawn([\amp, 0.9, \pan, 0, \sustain, 0.1]);
5.wait;
}
}
```

```

)
// 10秒の面の直前(0.05~0.5秒前)に1つの点を鳴らす
(
var d;
x.source = {
  loop {
    ~points.spawn([\amp, 0.9, \pan, 0, \sustain, 0.1]);
    d = rrand(0.05, 0.5).postln;
    d.wait;
    ~plane.spawn([\amp, 0.1, \pan, 0, \sustain, 10]);
    15.wait;
  }
}
)

```

面の前後に5つの点を鳴らしてみる。

```

// 10秒の面の直後(0.1秒後)に0.1秒間隔(10[Hz])で5つの点を鳴らす
(
x.source = {
  loop {
    ~plane.spawn([\amp, 0.1, \pan, 0, \sustain, 10]);
    10.1.wait;
    ~points.spawn([\amp, 0.9, \freq, 10, \pan, 0, \sustain, 0.5]);
    5.wait;
  }
}
)

```

```

// 10秒の面の直前(0.1秒前)に0.1秒間隔(10[Hz])で5つの点を鳴らす
(
x.source = {
  loop {
    ~points.spawn([\amp, 0.9, \freq, 10, \pan, 0, \sustain, 0.5]);
    0.6.wait;
    ~plane.spawn([\amp, 0.1, \pan, 0, \sustain, 10]);
    15.wait;
  }
}
)

```

```

// 10秒の面の後しばらく(5秒)たってから0.1秒間隔(10[Hz])で5つの点を鳴らす
(
x.source = {
  loop {
    ~plane.spawn([\amp, 0.1, \pan, 0, \sustain, 10]);
    15.wait;
    ~points.spawn([\amp, 0.9, \freq, 10, \pan, 0, \sustain, 0.5]);
    3.wait;
  }
}
)

```

面と点を同時に鳴らしてみる。

```

// 非常に小さな10秒の面の上で、ランダムに5つの点が鳴る
(
var d, l;
x.source = {
  loop {
    l = 10.0;
    ~plane.spawn([\amp, 0.01, \pan, 0, \sustain, 10]);
    5.do {
      d = l.rand;
      l = l-d;
      d.postln;
      d.wait;
      ~points.spawn([\amp, 0.9, \freq, 1, \pan, 1.0.rand2, \sustain, 0.1]);
    };
    5.wait;
  }
}
)

```

```

// 面と点の長さや位置をずらす
(

```

```

var d, l;
x.source = {
  loop {
    l = 12.0.rand;
    ~plane.spawn([\amp, 0.01, \pan, 0, \sustain, 10.0.rand]);
    5.do {
      d = l.rand;
      l = l-d;
      d.postln;
      d.wait;
      ~points.spawn([\amp, 0.9, \freq, 1, \pan, 1.0.rand2, \sustain, 0.1]);
    };
    5.wait;
  }
}
)

x.stop;

```

▼エクササイズ：点/線/面によるリズム

リズムは音響的な時間における秩序であり構造である。

点と非常に短い線と面を使ってパターンをつくり、反復させることでリズムをデザインせよ。

まず最初に、1つの面と8つの点によるパターンから出発する。

```

// 短い1つの面とそれに続く8つの点によるパターン
(
var i = ([ \plane ] ++ Array.fill(8, \points)).postln;
var a = ([ 0.1 ] ++ Array.rand(8, 0.5, 0.9)).postln;
var d = ([ 0.1 ] ++ Array.fill(8, 0.2)).postln;
var p = ([ 0 ] ++ Array.rand(8, -1, 1)).postln;
~out = Pbind(
  \instrument, Pseq(i, inf),
  \amp, Pseq(a, inf),
  \freq, 1,
  \pan, Pseq(p, inf),
  \sustain, 0.01,
  \dur, Pseq(d, inf)
);
)

```

面のパターンと点のパターンを独立につくり、同時に反復させてみる。

```

// ランダムな8つの面と等間隔の8つの点のパターンを合成する
(
var d1 = Array.exprand(8, 0.01, 2).postln;
var d2 = Array.fill(8, d1.sum/8).postln;
var pt1, pt2;
pt1 = Pbind(
  \instrument, \plane,
  \amp, 0.1,
  \freq, 1,
  \pan, Pfunc({1.rand2}),
  \sustain, 0.01,
  \dur, Pseq(d1, inf)
);
pt2 = Pbind(
  \instrument, \points,
  \amp, 0.9,
  \freq, 1,
  \pan, Pfunc({1.rand2}),
  \sustain, 0.1,
  \dur, Pseq(d2, inf)
);
~out = Ppar([pt1, pt2]);
)

```

さらに短くて高い線のパターンを加える。

```
// ランダムな8つの線と面のパターンと等間隔の8つの点のパターンを合成する
(
var d1 = Array.exprand(8, 0.01, 3).postln;
var d2 = Array.exprand(8, 0.01, 3).postln;
var d3 = Array.fill(8, [d1.sum, d2.sum].choose/8).postln;
var pt1, pt2, pt3;
pt1 = Pbind(
  \instrument, \line,
  \amp, 0.1,
  \freq, Pfunc({rrand(8000, 12000)}),
  \pan, Pfunc({1.rand2}),
  \sustain, Pfunc({0.1.rand}),
  \dur, Pseq(d1, inf)
);
pt2 = Pbind(
  \instrument, \plane,
  \amp, 0.1,
  \pan, Pfunc({1.rand2}),
  \sustain, 0.01,
  \dur, Pseq(d2, inf)
);
pt3 = Pbind(
  \instrument, \points,
  \amp, 0.9,
  \freq, 1,
  \pan, Pfunc({1.rand2}),
  \sustain, 0.1,
  \dur, Pseq(d3, inf)
);
~out = Ppar([pt1, pt2, pt3]);
)
```

最後に低い線のパターンを加える。合計4つのパターンが並列に反復されている。

```
// ランダムな8つの高い線と面、2つの低い線のパターンと等間隔の8つの点のパターンを合成する
(
var d1 = Array.exprand(8, 0.01, 3).postln;
var d2 = Array.exprand(8, 0.01, 3).postln;
var d3 = Array.fill(8, [d1.sum, d2.sum].choose/8).postln;
var d4 = Array.fill(2, [d1.sum, d2.sum].choose/2).postln;
var pt1, pt2, pt3, pt4;
pt1 = Pbind(
  \instrument, \line,
  \amp, 0.1,
  \freq, Pfunc({rrand(8000, 12000)}),
  \pan, Pfunc({1.rand2}),
  \sustain, Pfunc({0.1.rand}),
  \dur, Pseq(d1, inf)
);
pt2 = Pbind(
  \instrument, \plane,
  \amp, 0.1,
  \pan, Pfunc({1.rand2}),
  \sustain, 0.01,
  \dur, Pseq(d2, inf)
);
pt3 = Pbind(
  \instrument, \points,
  \amp, 0.9,
  \freq, 1,
  \pan, Pfunc({1.rand2}),
  \sustain, 0.1,
  \dur, Pseq(d3, inf)
);
pt4 = Pbind(
  \instrument, \line,
  \amp, 0.5,
  \freq, 50,
  \pan, 0,
  \sustain, 0.3,
  \dur, Pseq(d4, inf)
);
);
```

```
~out = Ppar([pt1, pt2, pt3, pt4]);  
)
```