

```
//Olyvea Davis - FINAL PROJECT:
```

```
/*Hi Dr. Rice! I just wanted to let you know how I set up my project. The first section (labelled "SYNTHDEFS CREATED," has all of my SynthDefs set up for later. After those are finished, I made another section called "ACTUAL SOUNDS PLAYED" that has all of the functions that will create sound. This made it easier for me to record the sounds in the order and with the timing that I wanted. Also, I tried grouping all of the SynthDefs together in order to boot them so that I wouldn't have to do each individually, but for some reason when I do that, SynthDef 2 gets mad at me for putting parenthesis at the beginning/end of it, so I had to just configure them one by one. Thanks and I hope you enjoy!*/
```

```
//Have a line of code that boots the audio server
```

```
s.boot;
```

```
//Set your sample rate to 44,100
```

```
s.options.sampleRate = 44100;
```

```
//Include a line of code that sets the audio output to the default speakers of your computer (keep in mind this might not work on my computer, but I want to see you type it)
```

```
s.options.outDevice = "Realtek High Definition Audio";
```

```
//The length of the project is somewhat arbitrary - 1.5 to 3 minutes in length
```

```
//Include detailed comments explaining your code
```

```
//Describe what each SynthDef is doing and why it works the way that it does
```

```

//SYNTHDEFS CREATED:

//Contains at least 15 unique SynthDefs

(//SynthDef 1

SynthDef.new(\soundOne, {
    arg freqOne=300;
    var sigOne, envOne, tempOne;
    //Each SynthDef must contain at least 1 EnvGen
    envOne = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
        //Env.new(levels, times, curve, doneAction)
        Env.new([0, 2, 0.8, 0], [2, 2, 3], [1, -1, 1]), doneAction:2);
    sigOne = 0;
    //At least 5 SynthDefs must contain iterative synthesis attained via the “.do” method
    10.do{ //DO 1 of 5
        //You must use at least 5 different wave-generating Ugens (i.e. SinOsc, Saw, Pulse, etc.)
        tempOne = SinOsc.ar(freqOne * //UGen 1 of 5
            {Rand(0.99, 1.02)},
            {Rand(0.0, 1.0)}, //gets rid of click at beginning of sound
            {ExpRand(0.005, 0.05)}; //chooses random duty cycle between 0.005 and 0.05
        );
        tempOne = tempOne * envOne; //use envOne to control the amplitude of tempOne
        sigOne = sigOne + tempOne;
    };
    //Each SynthDef creates stereo audio

```

```
sigOne = Splay.ar(sigOne) * 0.8; //some of tones might be going to speakers not available on device - splay fixes this and puts channels into stereo signal
Out.ar(0, sigOne);
Out.ar(1, sigOne);
}).add;
)
```

```
//SynthDef 2
```

```
SynthDef.new(\soundTwo, {
  arg freqTwo=50;
  var sigTwo, envTwo, sumTwo, tempTwo, reverbTwo;
  envTwo = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
    //Env.new(levels, times, loopNode, offset)
    Env.new([0, 0.7, 0.4, 0], [1, 2, 2], [-2, -1, 1]), doneAction:2);
  sigTwo = SinOsc.ar(freqTwo);
  //At least 10 SynthDefs must contain a Ugen being controlled by another Ugen
  //FreeVerb.ar(in, mix, room, damp, mul, add)
  reverbTwo = FreeVerb.ar(sigTwo, 0.7, 0.1); //Ugen Control 1 of 10
  10.do{ //DO 2 of 5
    tempTwo = FSinOsc.ar(freqTwo * //UGen 2 of 5
      {Rand(0.99, 1.02)}!2,
      {Rand(0.0, 1.0)}!2, //gets rid of click at beginning of sound
      {0.05}); //chooses random duty cycle between 0.005 and 0.05
  );
```

```
        sigTwo = sigTwo + tempTwo;

};

sigTwo = sigTwo * envTwo;

//Each SynthDef creates stereo audio

sigTwo = Splay.ar(sigTwo) * 0.8;

Out.ar(0, sigTwo);

Out.ar(1, sigTwo);

}).add;

)
```

```
(//SynthDef 3
```

```
SynthDef.new(\soundThree, {

    var sigThree, envThree, tempThree, freqThree;

    envThree = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
        //Env.new(levels, times, loopNode, offset)
        Env.new([0, 1, 1, 0], [2, 3, 2], [1, -2, 1]), doneAction:2);

    sigThree = 0;

    freqThree = ExpRand(50, 150);

    10.do{ //DO 3 of 5

        tempThree = FSinOsc.ar(freqThree * //Ugen Control 2 of 10
            {Rand(0.99, 1.02)},
            {Rand(0.0, 1.0)}, //gets rid of click at beginning of sound
            {0.05}); //chooses random duty cycle between 0.005 and 0.05
```

```

    );
    sigThree = sigThree + tempThree;
};
sigThree = sigThree * envThree;
//Each SynthDef creates stereo audio
sigThree = Splay.ar(sigThree) * 0.8;
Out.ar(0, sigThree);
Out.ar(1, sigThree);
}).add;
)

```

//SynthDef 4

```

SynthDef.new(\soundFour, {
    arg t_changeFour=0.5;
    var sigFour, envFour, freqFour, noiseFour, cutoffFour;
    envFour = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
        Env.new([0, 0.8, 0], [3, 4], [2, -2]), doneAction:2);
    noiseFour = LFNoise1.kr(2).exprange(50, 150); //UGen 3 of 5
    //In LFNoise.kr, the first # is how many times/sec frequency will change
    freqFour = LFNoise0.kr(noiseFour).exprange(50, 400); //Ugen Control 3 of 10
    cutoffFour = LFNoise1.kr(t_changeFour).exprange(400, 8000);
    //LFSaw.ar(freq, phase, mul, add)
    sigFour = LFSaw.ar(freqFour, 0, 0.25); //UGen 4 of 5

```

```

//FreeVerb.ar(signal, mix (aka wet vs dry), room size, damp)
sigFour = FreeVerb.ar(sigFour, 0.8, 0.2, 0.5);

//LPF (Low Pass Filter: filters that makes it sound farther away
sigFour = LPF.ar(in: sigFour, freq: cutoffFour, mul: 1, add: 0); //UGen 5 of 5
sigFour = sigFour * envFour;

//Each SynthDef creates stereo audio
sigFour = Splay.ar(sigFour) * 0.8;
Out.ar(0, sigFour);
Out.ar(1, sigFour);
}).add;
)

(//SynthDef 5
SynthDef.new(\soundFive, {
    arg fundFive=110, maxPartialFive=6, widthFive=0.5;
    var sigFive, envFive, ampFive, freqFive, waveFive;
    envFive = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
        //Env.new(levels, times, loopNode, offset)
        Env.new([0, 0.6, 0.4, 0], [1, 5, 4], [2, -3, -2]), doneAction:2);
    //XLine.kr(start, end, duration, mul, doneAction)
    ampFive = XLine.kr(ExpRand(10, 100), ExpRand(101, 200), 5);
    freqFive = LFNoise0.kr(4).exprange(fundFive, fundFive*maxPartialFive).round(fundFive);
    freqFive = freqFive * LFPulse.kr(6, add:1); //UGen Control 4 of 10

```

```

sigFive = Pulse.ar(freqFive, widthFive, ampFive);
//FreeVerb.ar(in, mix, room, damp, mul)
sigFive = FreeVerb.ar(sigFive, 0.5, 0.8, 0.25);
sigFive = sigFive * envFive;
//Each SynthDef creates stereo audio
sigFive = Splay.ar(sigFive) * 0.8;
Out.ar(0, sigFive);
Out.ar(1, sigFive);
}).add;
)

```

//SynthDef 6

```

SynthDef.new(\soundSix, {
  arg freqSix=200;
  var sigSix, envSix, waveSix;
  envSix = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
    //Env.new(levels, times, loopNode, offset)
    Env.new([0, 0.8, 0.3, 0.0], [4, 1.5, 2.0]), doneAction:2);
  sigSix = Pulse.ar(ExpRand(30, freqSix)) * envSix;
  freqSix = freqSix * envSix;
  sigSix = sigSix * FreeVerb.ar(sigSix, 0.8, 0.5, 0.2); //UGen Control 5 of 10
  //Each SynthDef creates stereo audio
  sigSix = Splay.ar(sigSix) * 0.8;

```

```

    Out.ar(0, sigSix);
    Out.ar(1, sigSix);
  }).add;
)

(//SynthDef 7
SynthDef.new(\soundSeven, {
  arg noiseSeven=130;
  var sigSeven, envSeven, waveSeven, freqSeven, soundEight;
  envSeven = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
    //Env.new(levels, times, loopNode, offset)
    Env.new([0, 0.2, 0.4, 0.6, 0.4, 0.2, 0.01], [1, 2, 3, 3, 2, 1], [-3, 2, -3, 2, -3, 2]), doneAction:2);
  waveSeven = XLine.ar(30, 210, 8);
  freqSeven = SinOsc.ar(ExpRand(noiseSeven, waveSeven)); //UGen Control 6 of 10
  sigSeven = LFTri.ar(150, 0.5) + freqSeven;
  sigSeven = FreeVerb.ar(sigSeven, 0.9, 0.5);
  sigSeven = sigSeven * envSeven;
  //Each SynthDef creates stereo audio
  sigSeven = Splay.ar(sigSeven) * 0.8;
  Out.ar(0, sigSeven);
  Out.ar(1, sigSeven);
}).add;
)

```



```

(//SynthDef 8
SynthDef.new(\soundEight, {
  arg freqEight=150;
  var sigEight, envEight, tempEight;
  envEight = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
    //Env.new(levels, times, loopNode, offset)
    Env.new([0, 0.8, 0], [4, 6], [1, -1]), doneAction:2);
  sigEight = 0;
  10.do{ //DO 4 of 5
    tempEight = FSinOsc.ar(freqEight *
      {Rand(0.98, 1.01)},
      {Rand(0.0, 0.9)}, //gets rid of click at beginning of sound
      {0.05}); //cycle
    );
    sigEight = sigEight + tempEight;
  };
  sigEight = LFTri.ar(freqEight) * sigEight; //UGen Control 7 of 10
  sigEight = sigEight * envEight;
  //Each SynthDef creates stereo audio
  sigEight = Splay.ar(sigEight) * 0.8;
  Out.ar(0, sigEight);
  Out.ar(1, sigEight);

```

```
}).add;  
)
```

```
(//SynthDef 9
```

```
SynthDef.new(\soundNine, {  
  var sigNine, envNine, ampNine;  
  envNine = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2  
    //Env.new(levels, times, loopNode, offset)  
    Env.new([0, 0.9, 0], [3, 6], [5, -2]), doneAction:2);  
  ampNine = SinOsc.kr((ExpRand(5, 20))!4).range(0, 0.9) * envNine; //UGen Control 8 of 10  
  sigNine = SinOsc.ar({ExpRand(50, 800)}!4); //match 4 random sine tones with 4 random frequencies  
  sigNine = sigNine * ampNine;  
  //Each SynthDef creates stereo audio  
  sigNine = Splay.ar(sigNine) * 0.7;  
  Out.ar(0, sigNine);  
  Out.ar(1, sigNine);  
}).add;  
)
```

```
(//SynthDef 10
```

```
SynthDef.new(\soundTen, {  
  var sigTen, envTen, sumTen, waveTen;
```

```

envTen = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
    //Env.new(levels, times, loopNode, offset)
    Env.new([0.01, 0.9, 0], [3, 4], [-3, -2]), doneAction:2);
sigTen = SinOsc.ar(ExpRand(110, 400)) * LFPulse.kr(6, add:1); //UGen Control 9 of 10
//FreeVerb.ar(in, mix, room, damp)
sigTen = sigTen * envTen;
//Each SynthDef creates stereo audio
sigTen = Splay.ar(sigTen);
Out.ar(0, sigTen);
Out.ar(1, sigTen);
}).add;
)

```

```

(//SynthDef 11
SynthDef.new(\soundEleven, {
    var sigEleven, envEleven, freqEleven, ampEleven;
    envEleven = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
        //Env.new(levels, times, loopNode, offset)
        Env.new([0, 0.9, 0.6, 0.9, 0.6, 0], [2, 1, 1, 1, 2], [1, -2, 1, -2, 1]), doneAction:2);
    freqEleven = ExpRand(50, 250);
    ampEleven = XLine.kr(ExpRand(50, 300), ExpRand(301, 500), 4);
    //LFPulse.kr(freq, phase, width, mul, add)
    sigEleven = LFPulse.kr(freqEleven, 0.5, 0.6, ampEleven, 1); //UGen Control 10 of 10

```

```

sigEleven = SinOsc.ar(sigEleven) * envEleven;
//Each SynthDef creates stereo audio
sigEleven = Splay.ar(sigEleven) * 0.9;
Out.ar(0, sigEleven);
Out.ar(1, sigEleven);
}).add;
)

(//SynthDef 12
SynthDef.new(\soundTwelve, {
  arg freqTwelve=100;
  var sigTwelve, envTwelve, tempTwelve;
  //freqTwelve = SinOsc.ar(ExpRand(100, 350));
  sigTwelve = (LFSaw.ar(310) + LFSaw.ar(280) + LFSaw.ar(230) + SinOsc.ar(372)) * 0.1;
  envTwelve = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
    //Env.new(levels, times, loopNode, offset)
    Env.new([0, 0.7, 0.6, 0], [2, 2, 2], [-2, -2, 2]), doneAction:2);
  10.do{ //DO 5 of 5
    tempTwelve = LFSaw.ar(freqTwelve *
      {Rand(0.97, 1.02)},
      {Rand(0.0, 0.9)}, //gets rid of click at beginning of sound
      {ExpRand(0.005, 0.05)}); //cycle
  };

```

```
        sigTwelve = sigTwelve + tempTwelve;
    };
    sigTwelve = sigTwelve * envTwelve;
    //Each SynthDef creates stereo audio
    sigTwelve = Splay.ar(sigTwelve) * 0.8;
    Out.ar(0, sigTwelve);
    Out.ar(1, sigTwelve);
}).add;
)
```

```
(//SynthDef 13
```

```
SynthDef.new(\soundThirteen, {
    var sigThirteen, envThirteen, freqThirteen;
    envThirteen = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2
        //Env.new(levels, times, loopNode, offset)
        Env.new([0, 0.7, 0], [15, 15], [1, 2]), doneAction:2);
    freqThirteen = LFTri.ar(260, 1);
    sigThirteen = SinOsc.ar(freqThirteen);
    //Each SynthDef creates stereo audio
    sigThirteen = Splay.ar(sigThirteen) * 0.6;
    Out.ar(0, sigThirteen);
    Out.ar(1, sigThirteen);
}).add;
```

)

```
//SynthDef 14
```

```
SynthDef.new(\soundFourteen, {  
  arg freqFourteen=260;  
  var sigFourteen, envFourteen, tempFourteen;  
  envFourteen = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2  
    //Env.new(levels, times, loopNode, offset)  
    Env.new([0.01, 0.7, 0.01], [2, 3], [-2, 2]), doneAction:2);  
  sigFourteen = SinOsc.ar(freqFourteen);  
  10.do{  
    tempFourteen = Pulse.ar(freqFourteen *  
      {Rand(0.98, 1.02)},  
      {Rand(0.0, 0.6)}, //gets rid of click at beginning of sound  
      {ExpRand(0.005, 0.05)}; //cycle  
  );  
  sigFourteen = sigFourteen + tempFourteen;  
};  
sigFourteen = LFTri.ar(sigFourteen);  
//Each SynthDef creates stereo audio  
sigFourteen = Splay.ar(sigFourteen) * 0.8;  
Out.ar(0, sigFourteen);  
Out.ar(1, sigFourteen);
```

```
}).add;  
)
```

```
(//SynthDef 15
```

```
SynthDef.new(\soundFifteen, {  
    arg freqFifteen=400;  
    var sigFifteen, envFifteen;  
    envFifteen = EnvGen.kr( //envelope, gate, levelScale, timeScale, doneAction:2  
        //Env.new(levels, times, loopNode, offset)  
        Env.new([0.01, 0.9, 0.6, 0.01], [3, 2, 2], [-3, 3, -3]), doneAction:2);  
    sigFifteen = PinkNoise.ar(0.4) * envFifteen;  
    //Each SynthDef creates stereo audio  
    sigFifteen = Splay.ar(sigFifteen) * 0.6;  
    Out.ar(0, sigFifteen);  
    Out.ar(1, sigFifteen);
```

```
}).add;  
)
```

```
/* _____ */
```

```
//Record the audio of your project as a .wav or .aiff file and include it in your submission
```

```
s.recSampleFormat = 24; //Do for recording sample
```

```
s.makeGui; //shows Gui pop up and can record
```

```
//Records file as .wav or .aiff
```

```
//ACTUAL SOUNDS PLAYED:
```

```
//You will create the piece by compiling various clumps of code in real time
```

```
//The piece should contain three general portions: Beginning, middle, and end
```

```
//BEGINNING:
```

```
(Pbind(\instrument, \soundOne,
```

```
  \dur, Pseq([0.2, 0.4, 0.6], 1), //can put # here (inf means play for infinity)
```

```
  \freqOne, Pseq([63, 64, 67].midicps, inf) //freq will not work unless it is an argument in the SynthDef
```

```
  //if # of times is different between dur and freq, will play as many times as the lowest number
```

```
  //if more events than dur steps, they just go in order and match up to dur steps (so sounds different a few times before going back to normal
```

```
).play;
```

```
)
```

```
(Pbind(\instrument, \soundOne,
```

```
  \dur, Pseq([0.2, 0.4, 0.6], 1),
```

```
  \freqOne, Pseq([57, 60, 63].midicps, inf)
```

```
).play;
```

```
)
```



```
(Pbind(\instrument, \soundOne,  
      \dur, Pseq([0.2, 0.4, 0.6, 0.1], 1),  
      \freqOne, Pseq([35, 60, 63, 67].midicps, inf)  
).play;  
)
```

```
(Pbind(\instrument, \soundOne,  
      \dur, Pseq([0.2, 0.4, 0.6, 0.1, 0.1], 1),  
      \freqOne, Pseq([35, 60, 67, 72, 74].midicps, inf)  
).play;  
)
```

```
(Pbind(\instrument, \soundTwo,  
      \dur, Pseq([0.2, 0.4, 0.6, 0.01], 1),  
      \freqTwo, Pseq([63, 64, 67, 35].midicps, inf)  
).play;  
)
```

```
(Pbind(\instrument, \soundTwo,  
      \dur, Pseq([0.2, 0.4, 0.6, 0.01], 1),  
      \freqTwo, Pseq([57, 60, 63, 35].midicps, inf)  
).play;  
)
```

```
(Pbind(\instrument, \soundTwo,  
      \dur, Pseq([0.2, 0.4, 0.6, 0.1], 1),  
      \freqTwo, Pseq([60, 63, 67, 35].midicps, inf)  
).play;  
)
```

```
(Pbind(\instrument, \soundTwo,  
      \dur, Pseq([0.2, 0.4, 0.6, 0.1, 0.1], 1),  
      \freqTwo, Pseq([60, 67, 72, 74, 35].midicps, inf)  
).play;  
)
```

```
(Pbind(\instrument, \soundThree,  
      \dur, Pseq([0.01, 0.01, 0.01, 0.01, 0.01], 2),  
      \freqThree, Pseq([60, 67, 72, 74, 35].midicps, inf)  
).play;  
)
```

```
//MIDDLE:
```

```
Synth.new(\soundSeven);
```

```
Synth.new(\soundFour);
```

```
Synth.new(\soundFour, \t_changeFour, 1);
```

```
Synth.new(\soundFive);
```

```
Synth.new(\soundFive);
```

```
Synth.new(\soundSix);
```

```
(Pbind(\instrument, \soundSix,
```

```
  \dur, Pseq([0.1, 0.2, 0.4, 0.1], 1),
```

```
  \freqSix, Pseq([52, 57, 60, 63].midicps, inf)
```

```
).play;
```

```
)
```

```
//BACKGROUND SOUND***
```

```
(e = Pbind(\instrument, \soundEight,
```

```
  \dur, Pseq([0.1], 650),
```

```
  \freqEight, Pseq([40, 43, 45].midicps, inf)
```

```
).play;
```

```
)
```

```
//***
```

```
(Pbind(\instrument, \soundEight,
```

```
  \dur, Pseq([0.1, 0.2, 0.4, 0.1], 1),
```

```
  \freqEight, Pseq([52, 57, 60, 63].midicps, inf)
```

```
).play;
```

```
)
```

```
(Pbind(\instrument, \soundEight,  
      \dur, Pseq([0.1, 0.2, 0.4, 0.1], 1),  
      \freqEight, Pseq([59, 51, 63, 67].midicps, inf)  
).play;
```

```
)
```

```
(Pbind(\instrument, \soundEight,  
      \dur, Pseq([0.1, 0.2, 0.4, 0.1], 1),  
      \freqEight, Pseq([52, 57, 60, 63].midicps, inf)  
).play;
```

```
)
```

```
(Pbind(\instrument, \soundEight,  
      \dur, Pseq([0.1, 0.2, 0.4, 0.1], 1),  
      \freqEight, Pseq([59, 51, 67, 72].midicps, inf)  
).play;
```

```
)
```

```
Synth.new(\soundNine);
```

```
Synth.new(\soundNine);
```

```
Synth.new(\soundTen);
```

```
Synth.new(\soundTen);
```

```
Synth.new(\soundTen);
```

```
Synth.new(\soundTen);
```

```
Synth.new(\soundEleven);
```

```
Synth.new(\soundEleven);
```

```
Synth.new(\soundTwelve);
```

```
(Pbind(\instrument, \soundTwelve,
```

```
  \dur, Pseq([0.2, 0.4, 0.6], 1),
```

```
  \freqTwelve, Pseq([63, 64, 67].midicps, inf)
```

```
).play;
```

```
)
```

```
(Pbind(\instrument, \soundTwelve,
```

```
  \dur, Pseq([0.2, 0.4, 0.6], 1),
```

```
  \freqTwelve, Pseq([57, 60, 63].midicps, inf)
```

```
).play;
```

```
)
```

```
(Pbind(\instrument, \soundTwelve,
```

```
  \dur, Pseq([0.2, 0.4, 0.6, 0.1], 1),
```

```
  \freqTwelve, Pseq([35, 60, 63, 67].midicps, inf)
```

```
).play;  
)
```

```
(Pbind(\instrument, \soundTwelve,  
      \dur, Pseq([0.2, 0.4, 0.6, 0.1, 0.1], 1),  
      \freqTwelve, Pseq([35, 60, 67, 72, 74].midicps, inf)
```

```
).play;  
)
```

```
//END:
```

```
//BACKGROUND SOUND***
```

```
Synth.new(\soundThirteen);
```

```
//***
```

```
(Pbind(\instrument, \soundFourteen,  
      \dur, Pseq([0.01, 0.01, 0.01, 0.4, 0.4], 1),  
      \freqFourteen, Pseq([48, 54, 60, 72, 74].midicps, inf)
```

```
).play;  
)
```

```
(Pbind(\instrument, \soundFourteen,  
      \dur, Pseq([0.2, 0.4, 0.6, 0.1, 0.1], 1),  
      \freqFourteen, Pseq([35, 60, 67, 72, 75].midicps, inf)
```

```
).play;
```

```
)
```

```
(Pbind(\instrument, \soundFourteen,
```

```
  \dur, Pseq([0.2, 0.4, 0.6, 0.1, 0.1], 1),
```

```
  \freqFourteen, Pseq([35, 60, 67, 72, 77].midicps, inf)
```

```
).play;
```

```
)
```

```
(Pbind(\instrument, \soundFourteen,
```

```
  \dur, Pseq([0.01], 1),
```

```
  \freqFourteen, Pseq([84].midicps, inf)
```

```
).play;
```

```
)
```

```
l = Synth.new(\soundFifteen);
```