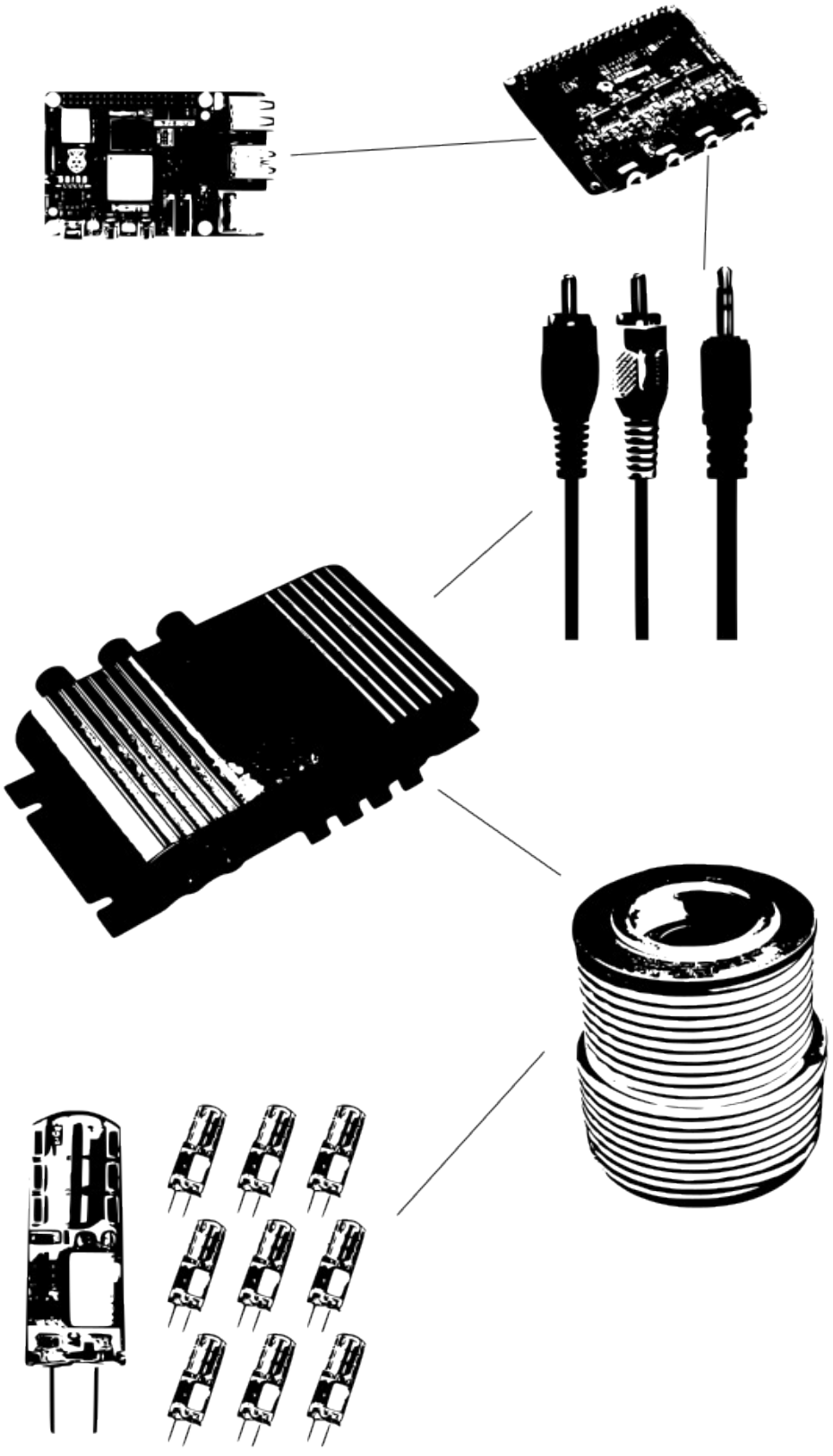

```
(
s.options.numWireBufs = 1024*16;
s.options.numOutputBusChannels=8;
s.options.numInputBusChannels=0;
TempoClock.default.tempo_(120/120);
s.options.memSize = 8192*64;
//s.options.blockSize = 64/4;
s.options.maxNodes = 1024 * 64;
s.waitForBoot
)

(
// 3D random
~a = VBAPSpeakerArray.new(3, [[180.rand2, 90.rand2], [180.rand2, 90.rand2], [180.rand2, 90.rand2],
[180.rand2, 90.rand2],[[180.rand2, 90.rand2], [180.rand2, 90.rand2], [180.rand2, 90.rand2], [180.rand2,
90.rand2]]);
~b = Buffer.loadCollection(s, ~a.getSetsAndMatrices);
)

(
SynthDef.new(\light, {
	arg  n1=1,
cutt=440,decay=6,t=2,attack=0.05,x=1,y=0.5,z=0.25,rx=1,ry=1,rz=1,rx1=1,ry1=1,rz1=1,ng1=1,ng2=1,ng3=1,
ng4=1,ng5=1,ng6=1,coef=0.1;
	var q,q1,i,j,k,i1,j1,k1,va,vb,v1,v2,v3,v4,v5,v6,ph_v1_x,or=0;
	var
ph_v1_y,ph_v1_z,ph_v2_x,ph_v2_y,ph_v2_z,ph_va_x,ph_va_y,ph_va_z,ph_vb_x,ph_vb_y,ph_vb_z;
	var osc1,osc2,osc3,freq, env1,env2,m=1,bank, pitch, fund, angle=0,angle1=90,r1,r2,p;
	var osc1_x,osc1_y,osc1_z,osc2_x,osc2_y,osc2_z,cvax,cvay,cvaz,cvbx,cvby,cvbz,cva,cvb,phx,phy;
	var spacex,timex,spacey,timey,spacez,timez,ang=(360-(360/1.6180339887499));
	var wrappedOut1 = { |busArray, signalArray|
	[busArray, signalArray].flop.do { |pair|
		Out.ar(pair[0], pair[1])}};
	var wrappedOut2 = { |busArray, signalArray|
	[busArray, signalArray].flop.do { |pair|
		Out.ar(pair[0], pair[1])}};

	var out = NamedControl.kr(\out, [0, 1, 2, 3,4,5,6,7]);  // multiple buses!

	r1=[rx,ry,rz].normalizeSum;
	r2=[rx1,ry1,rz1].normalizeSum;

	i = Quaternion(0, r1.[0].sqrt, 0, 0);
	j = Quaternion(0, 0,r1.[1].sqrt, 0);
	k = Quaternion(0, 0, 0, r1.[2].sqrt);

	i1 = Quaternion(0, r2.[0].sqrt, 0, 0);
	j1 = Quaternion(0, 0,r2.[1].sqrt, 0);
	k1 = Quaternion(0, 0, 0, r2.[2].sqrt);

	p=[x,y,z].normalizeSum;

	va=Quaternion(0,p.[0].sqrt*ng1,p.[1].sqrt*ng2,p.[2].sqrt*ng3);
	vb=Quaternion(0,p.[0].sqrt*ng4,p.[1].sqrt*ng5,p.[2].sqrt*ng6);


	q=(cos(ang.degrad/2)+(sin(ang.degrad/2)*(i+j+k)));
	q1=(cos(ang.degrad/2)+(sin(ang.degrad/2)*(i1+j1+k1)));

	va=q*va*q.conjugate;
	vb=q1*vb*q1.conjugate;
```

```
		cva=Cartesian(va.b-or,va.c-or,va.d-or);
		cvb=Cartesian(vb.b-or,vb.c-or,vb.d-or);

		or=or+coef;

env1=EnvGen.ar(Env.perc(attack,decay,pitch.reciprocal),doneAction:2,levelScale:phx.real.round(0.0001));
env2=EnvGen.ar(Env.perc(attack,decay,pitch.reciprocal),doneAction:2,levelScale:phx.imag.round(0.0001));


osc1_x=VBAP.ar(8,SinOsc.ar((220),phase:0.degrad),~b.bufnum,cva.theta.raddeg+LFSaw.ar(0.1,mul:180),cv
a.phi.raddeg+LFSaw.ar(0.1,mul:90));
osc1_y=VBAP.ar(8,SinOsc.ar((220),phase:90.degrad),~b.bufnum,cvb.theta.raddeg+LFSaw.ar(0.1,mul:180),c
vb.phi.raddeg+LFSaw.ar(0.1,mul:90);

m=m+1;

		wrappedOut1.value(out,LeakDC.ar((osc1_x)*env1));
		wrappedOut2.value(out, LeakDC.ar((osc1_y)*env2));

		//Out.ar(0,LPF.ar(LeakDC.ar((osc1_x)*env1),cutt));
		//Out.ar(0, LPF.ar(LeakDC.ar((osc1_y)*env2),cutt));

		};
}).add;
);

(
var  angle1=0,angle2=0,angle3=0,angle4=0,q1,q2,q3,q4,v1,v2,v3,v4,ph1,ph2,ph3,ph4,r,m=2;
var i = Quaternion(0, (1/3).sqrt, 0, 0),j = Quaternion(0, 0, (1/3).sqrt, 0),k = Quaternion(0, 0, 0, (1/3).sqrt);
var i1=0,i2=0,i3=0,i4=0;

r=[1/2,2/3,3/4].normalizeSum;

v1=Quaternion(0,r.[0].sqrt.neg,r.[1].sqrt,r.[2].sqrt);
v2=Quaternion(0,r.[0].sqrt.neg,r.[1].sqrt,r.[2].sqrt);
v3=Quaternion(0,r.[0].sqrt.neg,r.[1].sqrt,r.[2].sqrt);
v4=Quaternion(0,r.[0].sqrt.neg,r.[1].sqrt,r.[2].sqrt);

~axis=10000.collect{[0.999999.rand,0.999999.rand,0.999999.rand].normalizeSum;};

		~phase1 = 10000.collect{
			angle1=((360-(360/1.6180339887499)));
			q1=(cos(angle1.degrad/2)+(sin(angle1.degrad/2)*(i+j+k)));
		i=Quaternion(0,~axis.[i1].[0].sqrt,0,0);
		j=Quaternion(0,0,~axis.[i1].[1].sqrt,0);
		k=Quaternion(0,0,0,~axis.[i1].[2].sqrt);
		i1=i1+1;
		v1=q1*v1*q1.conjugate;
		ph1=(v1.a*m).round(0.25);
		};

~phase2 = 10000.collect{
			angle2=((360-(360/1.6180339887499)));
			q2=(cos(angle2.degrad/2)+(sin(angle2.degrad/2)*(i+j+k)));
		i=Quaternion(0,~axis.[i2].[0].sqrt,0,0);
		j=Quaternion(0,0,~axis.[i2].[1].sqrt,0);
		k=Quaternion(0,0,0,~axis.[i2].[2].sqrt);
		i2=i2+1;
		v2=q2*v2*q2.conjugate;
		ph2=(v2.b*m).round(0.25);
		};

~phase3 = 10000.collect{
```

```
                angle3=((360-(360/1.6180339887499)));
                q3=(cos(angle3.degrad/2)+(sin(angle3.degrad/2)*(i+j+k)));
        i=Quaternion(0,~axis.[i3].[0].sqrt,0,0);
        j=Quaternion(0,0,~axis.[i3].[1].sqrt,0);
        k=Quaternion(0,0,0,~axis.[i3].[2].sqrt);
        i3=i3+1;
        v3=q3*v3*q3.conjugate;
        ph3=(v3.c*m).round(0.25);

        };

~phase4 = 10000.collect{
                angle4=((360-(360/1.6180339887499)));
                q4=(cos(angle4.degrad/2)+(sin(angle4.degrad/2)*(i+j+k)));
        i=Quaternion(0,~axis.[i4].[0].sqrt,0,0);
        j=Quaternion(0,0,~axis.[i4].[1].sqrt,0);
        k=Quaternion(0,0,0,~axis.[i4].[2].sqrt);
        i4=i4+1;
        v4=q4*v4*q4.conjugate;
        ph4=(v4.d*m).round(0.25);
        };

);

(
b=Pdef(\2, Pbind(\instrument, \light,
        \dur,Pseq(~phase2.abs,inf),
                \out,Pn(Pshuf([0, 1, 2, 3,4,5,6,7], 1), inf).clump(4).collect([_]),
        \coef,Pexprand(0.0001,0.1,inf),
        \x,Pbrown(0,1,0.1,inf),
        \y,Pbrown(0,1,0.1,inf),
        \z,Pbrown(0,1,0.1,inf),
        \rx,Pbrown(0,1,0.1,inf),
        \ry,Pbrown(0,1,0.1,inf),
        \rz,Pbrown(0,1,0.1,inf),
                \rx1,Pbrown(0,1,0.1,inf),
        \ry1,Pbrown(0,1,0.1,inf),
        \rz1,Pbrown(0,1,0.1,inf),
                \ng1,Prand([1.neg,1,1.neg,1],inf),
        \ng2,Prand([1.neg,1,1.neg,1],inf),
        \ng3,Prand([1.neg,1,1.neg,1],inf),
                        \ng4,Prand([1.neg,1,1.neg,1],inf),
        \ng5,Prand([1.neg,1,1.neg,1],inf),
        \ng6,Prand([1.neg,1,1.neg,1],inf),
        \decay,Pbrown(2,8,1,inf);
));
)


(
b.play;
s.record(numChannels:8)

)
```