```
/* SC3 code used for the realtime composition of "OllieInDaTube"
based on a Messiaen-sample by L. Brusci <lorenzo.brusci@timet.org>
using JustInTime code Library by J. Rohrhuber <rohrhuber@uni-hamburg.de>
H. Hoelzl, Jan04 <hannes@earweego.net>
*/

p = ProxySpace.push;

/*
//record(key, path, headerFormat, sampleFormat)
t = p.record(\r, "OllieInDaTube.aiff");
t = ~r.record( "/Volumes/2/klaenge/iTunfische/OllieInDaTubeCarpet2.aiff");

// start recording
t.unpause;

// pause recording
t.pause;

// stop recording
t.close;
*/

// ~r = nice Combetc FX
~r.play;
~r.fadeTime = 6;
~r = { CombN.ar(~egg.ar, 0.01, [2.0.rand, 2.0.rand]*0.01, 1 )  }
~r = { CombL.ar(~egg.ar, 0.05, LFNoise0.kr(MouseY.kr*[3, 2], 0.4, 0.5).lag(0.4)*[2.0.rand,
2.0.rand]*0.05, 1, 0.3 )  }
~r = { CombL.ar(~egg.ar, 0.05, 1-~amp.kr *[1.0.rand, 1.0.rand].post * 0.05, 3, 0.3 )
};//TEPPICH
~r = { CombL.ar(~egg.ar, 0.05, 1-~amp.kr * Array.rand(4, 0.01, 1).post * 0.05, 1, 01.2 ).sum
};
~r = { BPF.ar(~egg.ar,  1-~amp.kr * 20000 * Array.rand(6, 0.01, 1).post, 0.1, 10  ).sum  }
~r = { RHPF.ar(~egg.ar,  1-~amp.kr ** 0.3 * 20000 * Array.exprand(2, 0.01, 1).post, 0.14, 8
)  }
// Iceberg Bass
~r = { CombL.ar(~egg.ar, 0.05, ~amp.kr ** 0.1 * Array.rand(2, 0.01, 1).post * 0.05, 10, 01.2
)  }
~r = { CombL.ar(~egg.ar, 0.05, 0.01 * Array.rand(2, 0.01, 1).post * 0.05, 115, 01.2 )  }
~r = { RLPF.ar(~egg.ar, ([100, 80.5]*Array.rand(4, 0.13, 0.9)).post, 0.01, 6 ).sum  }
~r.stop;


(
s.sendMsg(\b_allocRead, 1, "/Volumes/2/klaenge/iTunfische/Timet13Dino11.aif");
s.sendMsg(\b_allocRead, 2, "/Volumes/2/klaenge/iTunfische/TimetKMessiaenorch2.aif");
)

n = 1     //
n = 2     //
~egg.play;
~egg.fadeTime = 1.4;
~egg.stop
~egg = { BufRd.ar(2, n, Phasor.ar(0, BufRateScale.kr(n)*0.334, MouseY.kr * BufFrames.kr(n),
```

```
BufFrames.kr(n) * MouseX.kr)) * 1 };      // start-end-mouse
~egg = { BufRd.ar(2, n, LFSaw.ar(BufDur.ir(n).reciprocal).range(0, BufFrames.ir(n)) )
};//ORIG
~egg = { BufRd.ar(2, n, LFSaw.ar(BufDur.ir(n).reciprocal* 0.125).range(0, BufFrames.ir(n))
)*1 };//SLOw
~egg = { (BufRd.ar(2, n, LFSaw.ar(BufDur.ir(n).reciprocal * 0.0275).range(0,
BufFrames.ir(n)) ) - ~d.kr ).max(0) * 01.1 };//SLOw IceBergPeaks

~d = 0.15;           // ~d = Difference for Iceberg FX
~d = {SinOsc.kr(0.1, 0, 0.035, 0.1)};
~d = {SinOsc.kr(0.1, 0, 0.035, 0.1) * LFPulse.kr(8, 0.1, 1) };
~d = {LFNoise1.kr(10.1, 0.1, 0.15)};

                          // ~amp = Follower or sim
~amp = {Amplitude.kr(~egg.ar, 0.15, 0.002)};
~amp = { LFNoise0.kr(MouseY.kr*[13, 2], 0.4, 0.6).lag(0.6) };
```