

reflex

Claude Heiland-Allen

2009–2018

# Contents

|    |                                 |    |
|----|---------------------------------|----|
| 1  | Ball.hs . . . . .               | 3  |
| 2  | -/borgan~.pd . . . . .          | 3  |
| 3  | clap~.pd . . . . .              | 5  |
| 4  | cm-ead~.pd . . . . .            | 7  |
| 5  | .gitignore . . . . .            | 8  |
| 6  | handdrum~.pd . . . . .          | 8  |
| 7  | hihat~.pd . . . . .             | 10 |
| 8  | kick4~.pd . . . . .             | 11 |
| 9  | loopleftine~.pd . . . . .       | 12 |
| 10 | Makefile . . . . .              | 14 |
| 11 | -/pan4~.pd . . . . .            | 14 |
| 12 | pitchshift~.pd . . . . .        | 15 |
| 13 | Polychoron.hs . . . . .         | 16 |
| 14 | -/Polyhedron2.hs . . . . .      | 18 |
| 15 | -/Polyhedron.hs . . . . .       | 19 |
| 16 | -/ppmtodvd.sh . . . . .         | 20 |
| 17 | -/Quaternion.hs . . . . .       | 20 |
| 18 | README . . . . .                | 22 |
| 19 | reflex2.pd . . . . .            | 22 |
| 20 | -/reflex3d2d.hs . . . . .       | 34 |
| 21 | -/reflex-44100.sh . . . . .     | 38 |
| 22 | reflex.hs . . . . .             | 38 |
| 23 | -/reflexivity.pd . . . . .      | 46 |
| 24 | -/reflex-jack.sh . . . . .      | 46 |
| 25 | -/REFLEX-nano.pd . . . . .      | 46 |
| 26 | -/reflex-onescreen.sh . . . . . | 56 |
| 27 | -/reflex.pd . . . . .           | 56 |
| 28 | reflex.pd . . . . .             | 59 |
| 29 | -/REFLEX.pd . . . . .           | 69 |
| 30 | -/reflex.sh . . . . .           | 74 |
| 31 | -/rx-export~.pd . . . . .       | 75 |
| 32 | -/rx-kick~.pd . . . . .         | 75 |
| 33 | -/rx-move.pd . . . . .          | 76 |
| 34 | -/rx-ology.pd . . . . .         | 77 |
| 35 | -/rx-pack8.pd . . . . .         | 77 |
| 36 | -/rx-shaker~.pd . . . . .       | 78 |
| 37 | -/rx-shape.pd . . . . .         | 79 |
| 38 | -/rx-solid.pd . . . . .         | 80 |
| 39 | -/rx-visibility.pd . . . . .    | 81 |
| 40 | -/Snapshot.hs . . . . .         | 82 |
| 41 | Snapshot.hs . . . . .           | 83 |
| 42 | snare~.pd . . . . .             | 84 |

|    |                       |    |
|----|-----------------------|----|
| 43 | START.sh . . . . .    | 86 |
| 44 | ./Vector.hs . . . . . | 86 |
| 45 | Vector.hs . . . . .   | 92 |

## 1 Ball.hs

```
module Ball where
```

```
import Vector
```

```
5 ball :: R -> V3 -> M4
  ball rr v =
    let r = sqrt $ v `dot` v
        d = sqrt $ rr*rr + r*r
        c = rr / d
10    s = r / d
        V3 x y z = if r /= 0 then norm v else V3 0 0 0
    in M4
        (1-x*x*(1-c)) (- (1-c)*x*y) (- (1-c)*x*z) (s*x)
        (- (1-c)*x*y) (1-y*y*(1-c)) (- (1-c)*y*z) (s*y)
15    (- (1-c)*x*z) (- (1-c)*y*z) (1-z*z*(1-c)) (s*z)
        (-s*x) (-s*y) (-s*z) (c)
```

## 2 ./borgan~.pd

```
#N canvas 507 61 722 583 10;
#X obj 38 30 r pd;
#X obj 38 52 route dsp;
#X obj 38 73 select 1;
5 #X obj 132 129 samplerate~;
#X obj 132 177 t f f;
#X obj 11 10 loadbang;
#X obj 207 310 noise~;
#X obj 133 390 hip~ 10;
10 #X obj 131 433 lop~ 1000;
#X obj 131 454 lop~ 1000;
#X obj 11 194 metro;
#X obj 132 411 hip~ 10;
#X obj 41 150 /;
15 #X obj 11 96 t b b b;
#X text 24 567 borgan~ (copyleft) claud@mathr.co.uk;
#X obj 43 475 *~;
#X obj 504 296 0;
#X obj 42 129 1.024e+06;
20 #X obj 132 265 -~;
#X obj 132 367 expr~ tanh($v1*1.5);
#X obj 132 153 / 1024;
#X obj 131 541 table \ $0-osc-1 1027;
#X obj 132 220 tabosc4~ \ $0-osc-1;
25 #X obj 162 241 tabosc4~ \ $0-osc-1;
#X obj 132 517 tabwrite~ \ $0-osc-1;
#X obj 214 50 tabosc4~ \ $0-osc-1;
#X obj 282 541 table \ $0-osc-2 1027;
#X obj 344 50 tabosc4~ \ $0-osc-2;
30 #X obj 213 117 *~;
```

```
#X obj 343 117 *~;
#X obj 282 517 tabwrite~ \ $0-osc -2;
#X obj 378 174 f 0;
#X obj 413 173 + 1;
35 #X obj 414 195 mod 2;
#X obj 374 226 sel 0 1;
#X obj 273 99 vline~;
#X obj 359 97 -~;
#X obj 359 76 sig~ 1;
40 #X obj 282 472 delay 5;
#X obj 132 479 delay 5;
#X obj 162 300 -~;
#X obj 132 323 *~;
#X obj 142 343 *~;
45 #X obj 192 281 tabosc4~ \ $0-osc -2;
#X obj 162 260 tabosc4~ \ $0-osc -2;
#X obj 162 199 * -2;
#X obj 480 164 *~;
#X msg 494 96 0 10;
50 #X obj 495 123 vline~;
#X obj 544 101 vline~;
#X obj 544 78 mtof;
#X obj 479 203 outlet~;
#X msg 510 76 1 10;
55 #X obj 493 10 inlet;
#X obj 44 452 cos~;
#X obj 43 401 vline~;
#X msg 42 382 -0.25 \, 0.25 \ $1;
#X obj 40 350 f;
60 #X obj 207 332 *~ 0.1;
#X msg 273 77 \ $1 15;
#X obj 494 44 sel -1 0;
#X connect 0 0 1 0;
#X connect 1 0 2 0;
65 #X connect 2 0 13 0;
#X connect 3 0 12 1;
#X connect 3 0 20 0;
#X connect 4 0 22 0;
#X connect 4 0 44 0;
70 #X connect 4 1 45 0;
#X connect 5 0 13 0;
#X connect 6 0 58 0;
#X connect 7 0 11 0;
#X connect 8 0 9 0;
75 #X connect 9 0 15 1;
#X connect 10 0 31 0;
#X connect 11 0 8 0;
#X connect 12 0 10 1;
#X connect 12 0 57 1;
80 #X connect 13 0 10 0;
#X connect 13 1 17 0;
#X connect 13 2 3 0;
#X connect 15 0 24 0;
#X connect 15 0 30 0;
85 #X connect 16 0 22 1;
#X connect 16 0 23 1;
#X connect 16 0 43 1;
```

```

#X connect 16 0 44 1;
#X connect 17 0 12 0;
90 #X connect 18 0 41 0;
#X connect 19 0 7 0;
#X connect 20 0 4 0;
#X connect 22 0 18 0;
#X connect 23 0 18 1;
95 #X connect 25 0 28 0;
#X connect 27 0 29 0;
#X connect 28 0 46 0;
#X connect 29 0 46 0;
#X connect 31 0 32 0;
100 #X connect 31 0 34 0;
#X connect 31 0 59 0;
#X connect 32 0 33 0;
#X connect 33 0 31 1;
#X connect 34 0 39 0;
105 #X connect 34 1 38 0;
#X connect 35 0 28 1;
#X connect 35 0 36 1;
#X connect 35 0 41 1;
#X connect 36 0 29 1;
110 #X connect 36 0 42 1;
#X connect 37 0 36 0;
#X connect 38 0 30 0;
#X connect 38 0 57 0;
#X connect 38 0 16 0;
115 #X connect 39 0 24 0;
#X connect 39 0 57 0;
#X connect 39 0 16 0;
#X connect 40 0 42 0;
#X connect 41 0 19 0;
120 #X connect 42 0 19 0;
#X connect 43 0 40 1;
#X connect 44 0 40 0;
#X connect 45 0 23 0;
#X connect 45 0 43 0;
125 #X connect 46 0 51 0;
#X connect 47 0 48 0;
#X connect 48 0 46 1;
#X connect 49 0 25 0;
#X connect 49 0 27 0;
130 #X connect 50 0 49 0;
#X connect 52 0 48 0;
#X connect 53 0 60 0;
#X connect 54 0 15 0;
#X connect 55 0 54 0;
135 #X connect 56 0 55 0;
#X connect 57 0 56 0;
#X connect 58 0 19 0;
#X connect 59 0 35 0;
#X connect 60 0 47 0;
140 #X connect 60 2 50 0;
#X connect 60 2 52 0;

```

### 3 clap~.pd

```

#N canvas 0 0 545 467 10;
#X obj 262 80 noise~;
#X obj 336 325 *~;
#X obj 151 310 *~;
5 #X obj 167 250 *~ 0.8;
#X obj 148 343 *~ 0.2;
#X obj 376 145 del 30;
#X obj 377 165 del 20;
#X obj 378 185 del 15;
10 #X obj 376 206 del 10;
#X obj 63 217 *~ 0.1;
#X obj 85 147 bp~ 1300 10;
#X obj 253 152 bp~ 700 10;
#X obj 175 147 bp~ 2500 4;
15 #X obj 336 369 bp~ 2700 2;
#X obj 3 147 lop~ 8000;
#X obj 260 114 hip~ 300;
#X obj 322 11 inlet;
#X obj 339 415 outlet~;
20 #X obj 339 390 *~ 0.7;
#X obj 336 292 cm-ead~ 3 50;
#X obj 179 284 cm-ead~ 1 800;
#X obj 70 73 mtof;
#X obj 120 73 mtof;
25 #X obj 158 75 mtof;
#X obj 192 75 mtof;
#X obj 225 75 mtof;
#X obj 193 52 + 41;
#X obj 158 50 + 63;
30 #X obj 116 49 + 52;
#X obj 71 50 + 84;
#X obj 226 53 + 26;
#X obj 322 52 t b f;
#X obj 322 34 sel 0;
35 #X connect 0 0 14 0;
#X connect 0 0 15 0;
#X connect 1 0 13 0;
#X connect 2 0 1 1;
#X connect 2 0 4 0;
40 #X connect 3 0 2 1;
#X connect 4 0 13 0;
#X connect 5 0 6 0;
#X connect 5 0 19 0;
#X connect 6 0 7 0;
45 #X connect 6 0 19 0;
#X connect 7 0 8 0;
#X connect 7 0 19 0;
#X connect 8 0 19 0;
#X connect 9 0 3 0;
50 #X connect 10 0 3 0;
#X connect 11 0 3 0;
#X connect 12 0 3 0;
#X connect 13 0 18 0;
#X connect 14 0 9 0;
55 #X connect 15 0 10 0;
#X connect 15 0 11 0;
#X connect 15 0 12 0;

```

```

#X connect 16 0 32 0;
#X connect 18 0 17 0;
60 #X connect 19 0 1 0;
#X connect 20 0 2 0;
#X connect 21 0 14 1;
#X connect 22 0 10 1;
#X connect 23 0 12 1;
65 #X connect 24 0 11 1;
#X connect 25 0 15 1;
#X connect 26 0 24 0;
#X connect 27 0 23 0;
#X connect 28 0 22 0;
70 #X connect 29 0 21 0;
#X connect 30 0 25 0;
#X connect 31 0 5 0;
#X connect 31 0 19 0;
#X connect 31 0 20 0;
75 #X connect 31 1 30 0;
#X connect 31 1 26 0;
#X connect 31 1 27 0;
#X connect 31 1 28 0;
#X connect 31 1 29 0;
80 #X connect 32 1 31 0;

```

## 4 cm-ead~.pd

```

#N canvas 0 0 450 300 10;
#X obj 21 17 inlet;
#X obj 81 17 inlet;
#X obj 141 17 inlet;
5 #X obj 313 56 samplerate~;
#X obj 182 5 loadbang;
#X obj 182 32 t b b;
#X obj 141 68 f \$2;
#X obj 81 84 f \$1;
10 #X obj 97 201 vline~;
#X obj 66 157 f;
#X obj 126 159 f;
#X obj 81 137 clip 0 1;
#X obj 141 138 clip 0 1;
15 #X obj 44 237 rpole~;
#X obj 21 107 delay;
#X obj 21 202 vline~;
#X obj 21 155 f 0;
#X obj 44 262 outlet~;
20 #X obj 42 157 f 1;
#X obj 141 88 expr exp((1000*log(0.001))/($f1*$f2));
#X obj 81 118 expr exp((1000*log(0.001))/($f1*$f2));
#X obj 47 180 -;
#X obj 21 45 t b b b b b b;
25 #X connect 0 0 22 0;
#X connect 1 0 14 1;
#X connect 1 0 7 0;
#X connect 2 0 6 0;
#X connect 3 0 19 1;
30 #X connect 3 0 20 1;
#X connect 4 0 5 0;

```

```

#X connect 5 0 6 0;
#X connect 5 0 7 0;
#X connect 5 1 3 0;
35 #X connect 6 0 19 0;
#X connect 7 0 14 1;
#X connect 7 0 20 0;
#X connect 8 0 13 1;
#X connect 9 0 8 0;
40 #X connect 9 0 21 1;
#X connect 10 0 8 0;
#X connect 11 0 9 1;
#X connect 12 0 10 1;
#X connect 13 0 17 0;
45 #X connect 14 0 10 0;
#X connect 14 0 16 0;
#X connect 15 0 13 0;
#X connect 16 0 15 0;
#X connect 18 0 21 0;
50 #X connect 19 0 12 0;
#X connect 20 0 11 0;
#X connect 21 0 15 0;
#X connect 22 0 14 0;
#X connect 22 1 18 0;
55 #X connect 22 2 9 0;
#X connect 22 3 7 0;
#X connect 22 4 6 0;
#X connect 22 5 3 0;

```

## 5 .gitignore

```

*.o
*.hi
*.4d
cabal.sandbox.config
5 .cabal-sandbox
Polychoron
reflex

```

## 6 handdrum~.pd

```

#N canvas 0 0 610 531 10;
#X obj 121 192 osc~ 440;
#X obj 189 190 osc~ 440;
#X obj 254 189 osc~ 440;
5 #X obj 322 187 osc~ 440;
#X obj 122 219 *~;
#X obj 122 241 *~;
#X obj 123 263 *~;
#X obj 124 311 cos~;
10 #X obj 125 344 *~;
#X obj 39 290 *~;
#X msg 205 100 0;
#X obj 118 70 t b f;
#X obj 126 435 outlet~;
15 #X obj 120 4 inlet;
#X obj 121 133 * 0.8;
#X obj 189 132 * 1.6666;

```

```
#X obj 254 132 * 0.833;
#X obj 123 287 clip~ -1 1;
20 #X obj 121 165 + 7.88;
#X obj 189 163 + 8.66;
#X obj 322 161 + 7.88;
#X obj 254 162 + 4.725;
#X obj 126 410 *~ 0.4;
25 #X obj 205 286 *~ 0.15;
#X obj 81 370 clip~ -0.7 0.5;
#X msg 449 87 2.666666;
#X msg 388 87 1.14141;
#X obj 323 67 t b f;
30 #X obj 322 4 inlet;
#X obj 322 133 *;
#X obj 206 263 cm-ead~ 1 400;
#X obj 37 264 cm-ead~ 1 500;
#X obj 117 46 mtof;
35 #X obj 323 43 mtof;
#X obj 118 26 sel 0;
#X obj 324 23 sel 0;
#X connect 0 0 4 0;
#X connect 1 0 4 1;
40 #X connect 2 0 5 1;
#X connect 3 0 6 1;
#X connect 4 0 5 0;
#X connect 5 0 6 0;
#X connect 6 0 9 1;
45 #X connect 6 0 17 0;
#X connect 7 0 8 0;
#X connect 8 0 22 0;
#X connect 9 0 24 0;
#X connect 10 0 0 1;
50 #X connect 10 0 1 1;
#X connect 10 0 2 1;
#X connect 10 0 3 1;
#X connect 11 0 10 0;
#X connect 11 0 30 0;
55 #X connect 11 0 31 0;
#X connect 11 1 26 0;
#X connect 11 1 29 0;
#X connect 11 1 16 0;
#X connect 11 1 15 0;
60 #X connect 11 1 14 0;
#X connect 13 0 34 0;
#X connect 14 0 18 0;
#X connect 15 0 19 0;
#X connect 16 0 21 0;
65 #X connect 17 0 7 0;
#X connect 18 0 0 0;
#X connect 19 0 1 0;
#X connect 20 0 3 0;
#X connect 21 0 2 0;
70 #X connect 22 0 12 0;
#X connect 23 0 8 1;
#X connect 24 0 22 0;
#X connect 25 0 29 1;
#X connect 26 0 29 1;
```

```

75  #X connect 27 0 10 0;
    #X connect 27 0 30 0;
    #X connect 27 0 31 0;
    #X connect 27 1 25 0;
    #X connect 27 1 29 0;
80  #X connect 27 1 16 0;
    #X connect 27 1 15 0;
    #X connect 27 1 14 0;
    #X connect 28 0 35 0;
    #X connect 29 0 20 0;
85  #X connect 30 0 23 0;
    #X connect 31 0 9 0;
    #X connect 32 0 11 0;
    #X connect 33 0 27 0;
    #X connect 34 1 32 0;
90  #X connect 35 1 33 0;

```

## 7 hihat~.pd

```

#N canvas 0 0 459 480 10;
#X obj 140 7 inlet;
#X obj 202 400 outlet~;
#X obj 83 181 *~;
5  #X obj 230 183 *~;
    #X obj 150 213 *~;
    #X obj 139 265 +~;
    #X obj 150 300 osc~ 5435;
    #X obj 150 236 *~ 12000;
10  #X obj 23 144 osc~ 6221;
    #X obj 248 142 osc~ 2441;
    #X obj 228 235 noise~;
    #X obj 180 328 +~;
    #X obj 230 299 clip~ -0.7 0.7;
15  #X obj 180 354 *~;
    #X obj 235 352 *~;
    #X obj 245 373 *~ 0.06;
    #X obj 177 374 *~ 0.06;
    #X obj 78 238 sig~ 7435;
20  #X obj 228 256 bp~ 12000 12;
    #X obj 172 143 osc~ 3283;
    #X obj 98 144 osc~ 4109;
    #X obj 245 111 sqrt~;
    #X obj 246 87 cm-ead~ 2 800;
25  #X obj 139 89 cm-ead~ 2 200;
    #X obj 157 117 cm-ead~ 2 100;
    #X obj 139 65 t b f;
    #X obj 139 45 moses 48;
    #X obj 199 62 * 4;
30  #X obj 239 55 t b f;
    #X obj 321 64 * 8;
    #X obj 140 26 sel 0;
    #X connect 0 0 30 0;
    #X connect 2 0 4 0;
35  #X connect 3 0 4 1;
    #X connect 4 0 7 0;
    #X connect 5 0 6 0;
    #X connect 6 0 11 0;

```

```
40  #X connect 7 0 5 1;
    #X connect 8 0 2 0;
    #X connect 9 0 3 1;
    #X connect 10 0 18 0;
    #X connect 11 0 13 0;
    #X connect 11 0 14 0;
45  #X connect 12 0 11 1;
    #X connect 13 0 16 0;
    #X connect 14 0 15 0;
    #X connect 15 0 1 0;
    #X connect 16 0 1 0;
50  #X connect 17 0 5 0;
    #X connect 18 0 12 0;
    #X connect 19 0 3 0;
    #X connect 20 0 2 1;
    #X connect 21 0 13 1;
55  #X connect 22 0 21 0;
    #X connect 23 0 13 1;
    #X connect 24 0 14 1;
    #X connect 25 0 23 0;
    #X connect 25 0 24 0;
60  #X connect 25 1 27 0;
    #X connect 26 0 25 0;
    #X connect 26 1 28 0;
    #X connect 27 0 23 2;
    #X connect 28 0 22 0;
65  #X connect 28 0 24 0;
    #X connect 28 1 29 0;
    #X connect 29 0 22 2;
    #X connect 30 1 26 0;
```

## 8 kick4~.pd

```
    #N canvas 0 0 447 345 10;
    #X obj 30 26 inlet;
    #X obj 30 87 t b f;
    #X obj 30 157 vline~;
5  #X obj 30 185 *~;
    #X obj 30 215 *~;
    #X obj 30 263 ~ 0.25;
    #X obj 29 289 cos~;
    #X obj 30 240 *~;
10 #X obj 81 159 vline~;
    #X obj 75 25 loadbang;
    #X obj 75 109 sqrt;
    #X obj 30 110 f;
    #X msg 75 81 0.5;
15 #X obj 29 316 outlet~;
    #X obj 30 57 sel 0;
    #X msg 30 134 \"$1 \, 0 111;
    #X connect 0 0 14 0;
    #X connect 1 0 11 0;
20 #X connect 1 1 8 0;
    #X connect 2 0 3 0;
    #X connect 2 0 3 1;
    #X connect 3 0 4 0;
    #X connect 3 0 4 1;
```

```

25  #X connect 4 0 7 0;
    #X connect 5 0 6 0;
    #X connect 6 0 13 0;
    #X connect 7 0 5 0;
    #X connect 8 0 7 1;
30  #X connect 9 0 12 0;
    #X connect 10 0 11 1;
    #X connect 11 0 15 0;
    #X connect 12 0 10 0;
    #X connect 14 1 1 0;
35  #X connect 15 0 2 0;

```

## 9 loopline~.pd

```

#N canvas 173 62 450 580 10;
#X obj 101 322 delwrite~ \l 10000;
#X obj 206 300 delwrite~ \r 10000;
#X obj 115 227 limiter~ 50 1;
5  #X obj 219 252 z~ 64;
    #X obj 101 250 z~ 64;
    #X obj 101 274 *~;
    #X obj 205 273 *~;
    #X obj 101 202 hip~ 1;
10  #X obj 218 203 hip~ 1;
    #X obj 105 -39 vd~ \l;
    #X obj 216 -38 vd~ \r;
    #X obj 164 -88 sig~ 444;
    #X obj 83 -87 noise~;
15  #X obj 231 -85 noise~;
    #X obj 217 9 vcf~ 10;
    #X obj 216 160 ~ 0.25;
    #X obj 217 181 cos~;
    #X obj 103 8 vcf~ 10;
20  #X obj 102 162 ~ 0.25;
    #X obj 102 181 cos~;
    #X obj 82 344 outlet~;
    #X obj 182 344 outlet~;
    #X obj 296 -85 inlet;
25  #X obj 231 -61 *~ 0.001;
    #X obj 84 -66 *~ 0.001;
    #X obj 165 -67 *~ 4;
    #X obj 253 36 vcf~ 10;
    #X obj 143 35 vcf~ 10;
30  #X obj 161 10 *~ 2;
    #X obj 255 94 expr~ tanh($v1);
    #X obj 254 57 *~;
    #X obj 341 31 vline~;
    #X obj 145 94 expr~ tanh($v1);
35  #X obj 145 56 *~;
    #X obj 104 -14 *~ 1;
    #X obj 217 -13 *~ 1;
    #X obj 215 74 expr~ tanh($v1);
    #X obj 216 54 *~;
40  #X obj 104 76 expr~ tanh($v1);
    #X obj 104 54 *~;
    #X obj 297 13 vline~;
    #X obj 161 -23 sig~ 50;

```

```
#X obj 296 -7 route bass treble boost;
45 #X obj 386 101 vline~;
#X obj 215 135 *~;
#X obj 103 138 *~;
#X connect 2 0 5 1;
#X connect 2 0 6 0;
50 #X connect 3 0 6 1;
#X connect 4 0 5 0;
#X connect 5 0 0 0;
#X connect 5 0 20 0;
#X connect 6 0 1 0;
55 #X connect 6 0 21 0;
#X connect 7 0 4 0;
#X connect 7 0 2 0;
#X connect 8 0 3 0;
#X connect 8 0 2 1;
60 #X connect 9 0 34 0;
#X connect 10 0 35 0;
#X connect 11 0 25 0;
#X connect 12 0 24 0;
#X connect 13 0 23 0;
65 #X connect 14 0 37 0;
#X connect 15 0 16 0;
#X connect 16 0 8 0;
#X connect 17 0 39 0;
#X connect 18 0 19 0;
70 #X connect 19 0 7 0;
#X connect 22 0 42 0;
#X connect 23 0 35 0;
#X connect 24 0 34 0;
#X connect 25 0 9 0;
75 #X connect 25 0 10 0;
#X connect 26 0 30 0;
#X connect 27 0 33 0;
#X connect 28 0 27 1;
#X connect 28 0 26 1;
80 #X connect 29 0 44 0;
#X connect 30 0 29 0;
#X connect 31 0 30 1;
#X connect 31 0 33 1;
#X connect 32 0 45 0;
85 #X connect 33 0 32 0;
#X connect 34 0 17 0;
#X connect 34 0 27 0;
#X connect 35 0 14 0;
#X connect 35 0 26 0;
90 #X connect 36 0 44 0;
#X connect 37 0 36 0;
#X connect 38 0 45 0;
#X connect 39 0 38 0;
#X connect 40 0 39 1;
95 #X connect 40 0 37 1;
#X connect 41 0 17 1;
#X connect 41 0 14 1;
#X connect 41 0 28 0;
#X connect 42 0 40 0;
100 #X connect 42 1 31 0;
```

```

#X connect 42 2 43 0;
#X connect 43 0 44 1;
#X connect 43 0 45 1;
#X connect 44 0 15 0;
105 #X connect 45 0 18 0;

```

## 10 Makefile

```

all: reflex 3_4_3.4d

3_3_3.4d 3_4_3.4d 4_3_3.4d 5_3_3.4d: Polychoron
    ./Polychoron

5 Polychoron: Polychoron.hs Vector.hs .cabal-sandbox
    cabal exec -- ghc -O2 -Wall --make Polychoron.hs

reflex: reflex.hs Ball.hs Vector.hs Snapshot.hs .cabal-sandbox
10    cabal exec -- ghc -O2 -Wall --make reflex.hs

.cabal-sandbox:
    cabal sandbox init
    cabal install GLUT

```

## 11 \_/pan4~.pd

```

#N canvas 359 161 450 300 10;
#X obj 16 17 inlet ~;
#X obj 103 11 inlet ~;
#X obj 16 261 outlet ~;
5 #X obj 87 262 outlet ~;
#X obj 147 262 outlet ~;
#X obj 207 262 outlet ~;
#X obj 15 229 *~;
#X obj 86 229 *~;
10 #X obj 146 229 *~;
#X obj 206 233 *~;
#X obj 30 149 cos ~;
#X obj 100 169 cos ~;
#X obj 160 189 cos ~;
15 #X obj 219 185 cos ~;
#X obj 102 103 -~ 0.25;
#X obj 159 105 -~ 0.5;
#X obj 247 108 -~ 0.75;
#X obj 103 32 *~ 0.25;
20 #X obj 103 75 wrap ~;
#X obj 103 53 +~ 10;
#X obj 30 173 max~ 0;
#X obj 101 197 max~ 0;
#X obj 161 208 max~ 0;
25 #X obj 220 207 max~ 0;
#X connect 0 0 6 0;
#X connect 0 0 7 0;
#X connect 0 0 8 0;
#X connect 0 0 9 0;
30 #X connect 1 0 17 0;
#X connect 6 0 2 0;

```

```

#X connect 7 0 3 0;
#X connect 8 0 4 0;
#X connect 9 0 5 0;
35 #X connect 10 0 20 0;
#X connect 11 0 21 0;
#X connect 12 0 22 0;
#X connect 13 0 23 0;
#X connect 14 0 11 0;
40 #X connect 15 0 12 0;
#X connect 16 0 13 0;
#X connect 17 0 19 0;
#X connect 18 0 16 0;
#X connect 18 0 15 0;
45 #X connect 18 0 14 0;
#X connect 18 0 10 0;
#X connect 19 0 18 0;
#X connect 20 0 6 1;
#X connect 21 0 7 1;
50 #X connect 22 0 8 1;
#X connect 23 0 9 1;

```

## 12 pitchshift~.pd

```

#N canvas 77 55 502 515 10;
#X obj 113 278 cos~;
#X obj 113 305 *~;
#X obj 113 334 +~;
5 #X obj 282 196 wrap~;
#X obj 282 288 cos~;
#X obj 282 315 *~;
#X obj 282 169 +~ 0.5;
#X obj 113 226 -~ 0.5;
10 #X obj 113 252 *~ 0.5;
#X obj 282 227 -~ 0.5;
#X obj 282 260 *~ 0.5;
#X obj 114 110 phasor~;
#X obj 115 59 inlet~;
15 #X obj 12 34 delwrite~ \ $0-del 5000;
#X obj 345 290 vd~ \ $0-del;
#X obj 176 278 vd~ \ $0-del;
#X obj 112 361 outlet~;
#X obj 13 7 inlet~;
20 #X obj 345 228 *~ 100;
#X obj 176 226 *~ 100;
#X obj 345 261 +~ 2;
#X obj 176 252 +~ 2;
#X obj 114 84 expr~ (1-exp($v1*0.05776))*10;
25 #X connect 0 0 1 0;
#X connect 1 0 2 0;
#X connect 2 0 16 0;
#X connect 3 0 9 0;
#X connect 3 0 18 0;
30 #X connect 4 0 5 0;
#X connect 5 0 2 1;
#X connect 6 0 3 0;
#X connect 7 0 8 0;
#X connect 8 0 0 0;

```

```

35  #X connect 9 0 10 0;
    #X connect 10 0 4 0;
    #X connect 11 0 7 0;
    #X connect 11 0 6 0;
    #X connect 11 0 19 0;
40  #X connect 12 0 22 0;
    #X connect 14 0 5 1;
    #X connect 15 0 1 1;
    #X connect 17 0 13 0;
    #X connect 18 0 20 0;
45  #X connect 19 0 21 0;
    #X connect 20 0 14 0;
    #X connect 21 0 15 0;
    #X connect 22 0 11 0;

```

### 13 Polychoron.hs

```

module Main(main) where

import Control.Monad(liftM2)
import Data.List(deleteFirstsBy, nubBy)
5  import System.IO(withBinaryFile, hPutBuf, IOMode(WriteMode))
import Foreign.Storable(sizeOf)
import Foreign.Marshal.Array(withArrayLen)

import Vector

10 {-

Compute the fundamental region of the regular polytope {p,q,r}.

15 The result is the normal vectors of the bounding planes of the
fundamental region.

-}
fr4 :: Int -> Int -> Int -> [V4]
20 fr4 p q r =
    let v0 = V4 1 0 0 0
        v1 = m1 ^^*^ v0
        v2 = (m1 ^^*^ m2) ^^*^ v0
        v3 = ((m1 ^^*^ m2) ^^*^ m3) ^^*^ v0
25 m1 = M4 c1 s1 0 0 (-s1) c1 0 0 0 0 1 0 0 0 0 1
m2 = M4 c2 0 s2 0 0 1 0 0 (-s2) 0 c2 0 0 0 0 1
m3 = M4 c3 0 0 s3 0 1 0 0 0 0 1 0 (-s3) 0 0 c3
c1 = r1 / r0
c2 = r2 / r1
30 c3 = r3 / r2
s1 = sqrt (1 - c1*c1)
s2 = sqrt (1 - c2*c2)
s3 = sqrt (1 - c3*c3)
r0 = radius 0 [p,q,r]
35 r1 = radius 1 [p,q,r]
r2 = radius 2 [p,q,r]
r3 = radius 3 [p,q,r]
    in [ norm $ cross4 v1 v2 v3
        , norm $ cross4 v0 v2 v3 ^* (-1)
40   , norm $ cross4 v0 v1 v3

```

```

    , norm $ cross4 v0 v1 v2 ^* (-1) ]

{-
45 Magic delta determinant used for calculating radii of sub-polytopes. These
    radii can be used to find the angles subtended by sub-polytopes, for example.

-}

50 delta :: [Int] -> R
    delta [] = 1
    delta [p] = let s = sin (pi / fromIntegral p) in s * s
    delta (p:q:rs) = let c = cos (pi / fromIntegral p) in delta (q:rs) - delta rs * ↵
        ↵ c * c

55 {-
    Compute the squared radius of a sub-polytope.

-}

60 radius2 :: Int -> [Int] -> R
    radius2 0 [] = 1
    radius2 0 (p:qs) = delta qs / delta (p:qs)
    radius2 j ps = radius2 0 ps - radius2 0 (take (j-1) ps)

65 {-
    Non-squared radius.

70 -}

    radius :: Int -> [Int] -> R
    radius n ps = sqrt $ radius2 n ps

75 {-
    Compute the transitive closure of a set of generators.

    In closure' gs ms ns:
80     gs : the generators
        ms : the accumulated closure
        ns : the new elements added last iteration

-}

85 closure :: (Approx m, M m) => [m] -> [m]
    closure gs = closure' gs gs gs

    closure' :: (Approx m, M m) => [m] -> [m] -> [m] -> [m]
90 closure' gs ms ns =
    let new = (deleteFirstsBy (~=) . nubBy (~=) . liftM2 (^*^^) gs $ ns) ms
    in if null new then ms else closure' gs (ms ++ new) new

{-
95 A polytope is represented by the matrices that make up its symmetry group.

```

```
-}
```

```
100 polychoron :: Int -> Int -> Int -> [M4]
    polychoron p q r = closure . map reflector $ fr4 p q r

main = do
    mapM_ write4D [(3, 3, 3), (4, 3, 3), (3, 4, 3), (5, 3, 3)]
105   where
        write4D (p, q, r) =
            let name = show p ++ "_" ++ show q ++ "_" ++ show r ++ ".4d"
            in  withBinaryFile name WriteMode $ \h ->
                withArrayLen (concatMap toList $ polychoron p q r) $ \n ptr -> do
110                 putStrLn $ name ++ " (" ++ show n ++ ")"
                    hPutBuf h ptr (n * sizeof (undefined::R))
                toList (M4 a b c d e f g h i j k l m n o p) = [a,b,c,d,e,f,g,h,i,j,k,l,m,n,o ↵
                    ↵ ,p]
```

## 14 -/Polyhedron2.hs

```
module Main(main) where

import Control.Monad(liftM2)
import Data.List(deleteFirstsBy, nubBy)
5 import System.IO(withBinaryFile, hPutBuf, IOMode(WriteMode))
import Foreign.Storable(sizeof)
import Foreign.Marshal.Array(withArrayLen)

import Vector

10 fr3 :: Int -> Int -> [V3]
    fr3 p q =
        let v0 = V3 1 0 0
            v1 = m1 ^^*^^ v0
15            v2 = (m1 ^^*^^ m2) ^^*^^ v0
            m1 = M3 c1 s1 0 (-s1) c1 0 0 0 1
            m2 = M3 c2 0 s2 0 1 0 (-s2) 0 c2
            c1 = r1 / r0
            c2 = r2 / r1
20            s1 = sqrt (1 - c1*c1)
            s2 = sqrt (1 - c2*c2)
            r0 = radius 0 [p,q]
            r1 = radius 1 [p,q]
            r2 = radius 2 [p,q]
25        in [ norm $ cross3 v1 v2
            , norm $ cross3 v0 v2 ^* (-1)
            , norm $ cross3 v0 v1 ]

delta :: [Int] -> R
30 delta [] = 1
    delta [p] = let s = sin (pi / fromIntegral p) in s * s
    delta (p:q:rs) = let c = cos (pi / fromIntegral p) in delta (q:rs) - delta rs * ↵
        ↵ c * c

radius2 :: Int -> [Int] -> R
35 radius2 0 [] = 1
    radius2 0 (p:qs) = delta qs / delta (p:qs)
```

```

radius2 j ps = radius2 0 ps - radius2 0 (take (j-1) ps)

radius :: Int -> [Int] -> R
40 radius n ps = sqrt $ radius2 n ps

closure :: (Approx m, M m) => [m] -> [m]
closure gs = closure' gs gs gs

45 closure' :: (Approx m, M m) => [m] -> [m] -> [m] -> [m]
closure' gs ms ns =
  let new = (deleteFirstsBy (~=) . nubBy (~=) . liftM2 (^*^^) gs $ ns) ms
  in if null new then ms else closure' gs (ms ++ new) new

50 polyhedron :: Int -> Int -> [M3]
polyhedron p q = closure . map reflector $ fr3 p q

main = do
  mapM_ write3D [(3, 3), (4, 3), (5, 3)]
55 where
  write3D (p, q) =
    let name = "poly_" ++ show p ++ "_" ++ show q
    in putStrLn $ name ++ " = " ++ show (map toList . filter ((0 <) . det) $
      ↪ polyhedron p q) ++ ";"
  toList (M3 a b c d e f g h i) = [[a,b,c],[d,e,f],[g,h,i]]

```

## 15 -/Polyhedron.hs

```

module Main(main) where

import Control.Monad(liftM2)
import Data.List(deleteFirstsBy, nubBy)
5 import System.IO(withBinaryFile, hPutBuf, IOMode(WriteMode))
import Foreign.Storable(sizeOf)
import Foreign.Marshal.Array(withArrayLen)

import Vector

10 fr3 :: Int -> Int -> [V3]
fr3 p q =
  let v0 = V3 1 0 0
      v1 = m1 ^^*^ v0
15      v2 = (m1 ^^*^^ m2) ^^*^ v0
      m1 = M3 c1 s1 0 (-s1) c1 0 0 0 1
      m2 = M3 c2 0 s2 0 1 0 (-s2) 0 c2
      c1 = r1 / r0
      c2 = r2 / r1
20      s1 = sqrt (1 - c1*c1)
      s2 = sqrt (1 - c2*c2)
      r0 = radius 0 [p,q]
      r1 = radius 1 [p,q]
      r2 = radius 2 [p,q]
25 in [ norm $ cross3 v1 v2
      , norm $ cross3 v0 v2 ^* (-1)
      , norm $ cross3 v0 v1 ]

delta :: [Int] -> R
30 delta [] = 1

```

```

delta [p] = let s = sin (pi / fromIntegral p) in s * s
delta (p:q:rs) = let c = cos (pi / fromIntegral p) in delta (q:rs) - delta rs * ↵
    ↵ c * c

radius2 :: Int -> [Int] -> R
35 radius2 0 [] = 1
radius2 0 (p:qs) = delta qs / delta (p:qs)
radius2 j ps = radius2 0 ps - radius2 0 (take (j-1) ps)

radius :: Int -> [Int] -> R
40 radius n ps = sqrt $ radius2 n ps

closure :: (Approx m, M m) => [m] -> [m]
closure gs = closure' gs gs gs

45 closure' :: (Approx m, M m) => [m] -> [m] -> [m] -> [m]
closure' gs ms ns =
    let new = (deleteFirstsBy (~=) . nubBy (~=) . liftM2 (^^*) gs $ ns) ms
    in if null new then ms else closure' gs (ms ++ new) new

50 polyhedron :: Int -> Int -> [M3]
polyhedron p q = closure . map reflector $ fr3 p q

main = do
    mapM_ write3D [(3, 3), (4, 3), (5, 3)]
55 where
    write3D (p, q) =
        let name = show p ++ "_" ++ show q ++ ".3d"
        in withBinaryFile name WriteMode $ \h ->
            withArrayLen (concatMap toList $ polyhedron p q) $ \n ptr -> do
60                putStrLn $ name ++ " (" ++ show (n `div` (3*3)) ++ ")"
                hPutBuf h ptr (n * sizeof (undefined::R))
            toList (M3 a b c d e f g h i) = [a,b,c,d,e,f,g,h,i]

```

## 16 -/ppmtodvd.sh

```

#!/bin/bash
# usage: ./ppmtodvd.sh output.m2v < input.ppm
# assumes: input ppm stream is 788x576
ppmtoy4m -S 444 -F 25:1 |
5 y4mscaler -O preset=dvd -O yscale=1/1 |
mpeg2enc -f 8 -q 2 -b 8000 -P -R 2 -o "$1"

```

## 17 -/Quaternion.hs

```

module Quaternion where

import Vector(R, M4(M4))

5 data Q = Q !R !R !R !R

qnorm2 :: Q -> R
qnorm2 (Q a b c d) = a*a + b*b + c*c + d*d

10 qnorm :: Q -> R
qnorm = sqrt . qnorm2

```

```

qconj :: Q -> Q
qconj (Q a b c d) = Q a (-b) (-c) (-d)
15
qinv :: Q -> Q
qinv q = qconj q 'qmults' (1/qnorm2 q)

qadd :: Q -> Q -> Q
20 qadd (Q a b c d) (Q x y z w) = Q (a+x) (b+y) (c+z) (d+w)

qmul :: Q -> Q -> Q
qmul (Q q1 q2 q3 q4) (Q p1 p2 p3 p4) = Q r1 r2 r3 r4
  where
25   r1 = q1 * p1 - q2 * p2 - q3 * p3 - q4 * p4
   r2 = q2 * p1 + q1 * p2 - q4 * p3 + q3 * p4
   r3 = q3 * p1 + q4 * p2 + q1 * p3 - q2 * p4
   r4 = q4 * p1 - q3 * p2 + q2 * p3 + q1 * p4

30 qexp :: Q -> Q
qexp (Q w x y z) = Q (exp w * c) (x * s) (y * s) (z * s)
  where
   n = sqrt $ x*x + y*y + z*z
   c = cos n
35   s = sin n / n

qlog :: Q -> Q
qlog (Q w x y z) = Q (log qn) (x * sac) (y * sac) (z * sac)
  where
40   vn2 = x*x + y*y + z*z
   qn2 = vn2 + w*w
   vn = sqrt vn2
   qn = sqrt qn2
   sac = acos (w / qn) / vn
45

qmults :: Q -> R -> Q
qmults (Q w x y z) f = Q (w*f) (x*f) (y*f) (z*f)

qpows :: Q -> R -> Q
50 qpows q f = qexp $ qlog q 'qmults' f

qslerp :: Q -> Q -> R -> Q
qslerp p q t = ((q 'qmults' qinv p) 'qpows' t) 'qmults' p

55 mquat2 :: Q -> Q -> M4
mquat2 (Q q0 q1 q2 q3) (Q p0 p1 p2 p3) = M4
  ( q0*p0 + q1*p1 + q2*p2 + q3*p3)
  ( q1*p0 - q0*p1 - q3*p2 + q2*p3)
  ( q2*p0 + q3*p1 - q0*p2 - q1*p3)
60  ( q3*p0 - q2*p1 + q1*p2 - q0*p3)
  (- q1*p0 + q0*p1 - q3*p2 + q2*p3)
  ( q0*p0 + q1*p1 - q2*p2 - q3*p3)
  (- q3*p0 + q2*p1 + q1*p2 - q0*p3)
  ( q2*p0 + q3*p1 + q0*p2 + q1*p3)
65  (- q2*p0 + q3*p1 + q0*p2 - q1*p3)
  ( q3*p0 + q2*p1 + q1*p2 + q0*p3)
  ( q0*p0 - q1*p1 + q2*p2 - q3*p3)
  (- q1*p0 - q0*p1 + q3*p2 + q2*p3)

```

```

70      (- q3*p0 - q2*p1 + q1*p2 + q0*p3)
      (- q2*p0 + q3*p1 - q0*p2 + q1*p3)
      (  q1*p0 + q0*p1 + q3*p2 + q2*p3)
      (  q0*p0 - q1*p1 - q2*p2 + q3*p3)

```

## 18 README

reflex

Audiovisual performance exploring variously truncated 4D polytopes.

5

```

$ make
$ ./START.sh

```

Requires Fostex Mixtab MIDI controller on channel 15 or compatible.

## 19 reflex2.pd

```

#N canvas 0 0 1099 692 10;
#X obj 28 28 netreceive 6666;
#X obj 24 642 netsend;
#X msg 24 593 connect localhost 5555;
5 #X obj 101 517 list prepend send;
#X obj 101 539 list trim;
#X obj 24 571 spigot 1;
#X obj 107 615 == 0;
#X obj 42 527 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
10 1;
#X obj 24 506 loadbang;
#X obj 24 547 metro 1000;
#X obj 101 565 spigot 0;
#X obj 207 553 loadbang;
15 #X obj 206 580 delay 1000;
#X msg 207 607 \; pd dsp 1;
#X obj 256 36 ctlin;
#X obj 241 73 pack f f f;
#X msg 301 104 set \ $3 \ $2 \ $1;
20 #X msg 301 131 15 21 88;
#X obj 239 130 route 15;
#X obj 239 158 route 66 67 68 69 70 71 72 73;
#X obj 186 356 sel 0 1;
#X obj 182 213 >= 64;
25 #X msg 186 381 XY Off;
#X msg 205 405 XY On;
#X obj 186 439 t a;
#X msg 239 106 \ $3 \ $2 \ $1;
#X obj 246 356 sel 0 1;
30 #X obj 250 180 >= 64;
#X obj 246 439 t a;
#X obj 306 356 sel 0 1;
#X obj 290 183 >= 64;
#X obj 306 439 t a;
35 #X obj 366 356 sel 0 1;
#X obj 362 183 >= 64;
#X obj 366 439 t a;
#X obj 426 356 sel 0 1;

```

```

#X obj 381 203 >= 64;
40 #X obj 426 439 t a;
#X obj 486 356 sel 0 1;
#X obj 477 260 >= 64;
#X obj 486 439 t a;
#X msg 246 381 XZ Off;
45 #X msg 265 405 XZ On;
#X msg 306 381 XW Off;
#X msg 325 405 XW On;
#X msg 366 381 YZ Off;
#X msg 385 405 YZ On;
50 #X msg 426 381 YW Off;
#X msg 445 405 YW On;
#X msg 486 381 ZW Off;
#X msg 505 405 ZW On;
#X obj 184 482 t a;
55 #X obj 32 435 dac~;
#X obj 558 350 route 16 17 18 19;
#X obj 577 368 t b f;
#X obj 560 391 pack 1 1 1 1;
#X obj 618 366 t b f;
60 #X obj 659 369 t b f;
#X msg 563 421 Shape \$1 \$2 \$3 \$4;
#X obj 710 203 t b f;
#X obj 696 235 pack 1 1 1 1;
#X obj 750 203 t b f;
65 #X obj 790 203 t b f;
#X obj 692 146 route 20 21 22 23;
#X obj 697 265 unpack f f f f;
#X obj 696 293 *;
#X obj 726 293 *;
70 #X obj 756 293 *;
#X obj 701 346 pack f f f;
#X msg 698 400 Move \$1 \$2 \$3;
#X obj 695 175 - 64;
#X obj 727 176 - 64;
75 #X obj 765 175 - 64;
#X obj 698 316 / 1000;
#X obj 748 316 / 1000;
#X obj 798 316 / 1000;
#X msg 302 247 2 \$1;
80 #X msg 254 316 1 \$1;
#X msg 333 322 3 \$1;
#X msg 393 271 4 \$1;
#X msg 429 330 5 \$1;
#X msg 218 253 0 \$1;
85 #X obj 218 206 moses 64;
#X msg 217 226 0;
#X obj 263 227 - 64;
#X obj 254 274 moses 64;
#X msg 253 294 0;
90 #X obj 299 295 - 64;
#X obj 302 202 moses 64;
#X msg 301 222 0;
#X obj 347 223 - 64;
#X obj 334 274 moses 64;
95 #X msg 333 294 0;

```

```

#X obj 379 295 - 64;
#X obj 392 225 moses 64;
#X msg 391 245 0;
#X obj 437 246 - 64;
100 #X obj 432 285 moses 64;
#X msg 431 305 0;
#X obj 477 306 - 64;
#X obj 262 248 / 32;
#X obj 298 316 / 32;
105 #X obj 346 244 / 32;
#X obj 378 316 / 32;
#X obj 436 267 / 32;
#X obj 476 327 / 32;
#X obj 602 47 f 0;
110 #X obj 675 38 + 1;
#X obj 602 2 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0 1
;
#X obj 602 23 metro 222;
#X obj 675 59 mod 8;
115 #X obj 297 595 s \${0-controllermap};
#X obj 292 530 loadbang;
#X obj 331 59 table \${0-controllernow 128;
#X obj 333 36 s \${0-controllernow;
#X obj 581 488 until;
120 #X msg 581 467 128;
#X obj 581 513 f 0;
#X obj 638 504 + 1;
#X obj 638 528 mod 128;
#X obj 581 566 tabread \${0-controllernow;
125 #X obj 581 539 t f f;
#X obj 716 597 +;
#X obj 731 566 * 128;
#X obj 284 674 table \${0-controllers 1024;
#X obj 581 619 tabwrite \${0-controllers;
130 #X msg 286 640 const -1;
#X obj 286 658 s \${0-controllers;
#X obj 582 444 t b f;
#X obj 854 440 / 16;
#X obj 855 466 int;
135 #X obj 841 91 hradio 15 1 0 8 empty empty empty 0 -8 0 10 -262144 -1
-1 0;
#X obj 800 508 f;
#X obj 826 383 route 7 48;
#X obj 795 414 >= 64;
140 #X obj 795 434 change;
#X obj 796 457 sel 1;
#X obj 841 63 hradio 15 1 0 8 empty empty empty 0 -8 0 10 -262144 -1
-1 0;
#X obj 533 111 until;
145 #X msg 533 87 128;
#X obj 533 67 t b f;
#X obj 579 82 * 128;
#X obj 533 137 f 0;
#X obj 590 118 + 1;
150 #X obj 590 142 mod 128;
#X obj 533 163 t f f;
#X obj 562 183 tabread \${0-controllermap;

```

```

#X obj 531 207 spigot;
#X obj 532 230 t f f;
155 #X obj 533 252 +;
#X obj 535 271 tabread \ $0-controllers;
#X obj 535 292 moses -0.5;
#X obj 534 313 swap;
#X obj 534 334 pack f f;
160 #X obj 296 615 table \ $0-controllermap 128;
#X msg 298 555 const 0 \, 16 1 \, 17 1 \, 18 1 \, 19 1 \, 66 1 \, 67
1 \, 68 1 \, 69 1 \, 70 1 \, 71 1;
#N canvas 51 0 928 647 \ $0-audio 0;
#X obj 157 422 outlet~;
165 #X obj 192 400 outlet~;
#X obj 171 326 limiter~ 50 1;
#X obj 206 350 z~ 64;
#X obj 158 348 z~ 64;
#X obj 158 382 *~;
170 #X obj 191 381 *~;
#X obj 158 299 hip~ 1;
#X obj 207 301 hip~ 1;
#X obj 133 178 loopline~;
#X obj 79 16 inlet;
175 #X obj 17 128 / 127;
#X obj 96 131 / 127;
#X obj 179 136 * 2;
#X msg 179 157 boost \ $1 10;
#X msg 96 156 treble \ $1 10;
180 #X msg 17 156 bass \ $1 10;
#X obj 131 265 pitchshift~;
#X obj 211 265 pitchshift~;
#X obj 259 186 vline~;
#X obj 246 50 metro 111;
185 #X obj 247 22 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
1;
#X obj 278 70 + 1;
#X msg 256 165 0;
#X msg 281 164 3;
190 #X msg 304 163 -2;
#X obj 345 236 kick4~;
#X obj 345 205 f 64;
#X obj 338 56 - 4;
#X obj 337 121 max 0;
195 #X obj 337 82 div 4;
#X obj 337 102 * 2;
#X obj 370 59 / 127;
#X obj 378 114 * 8;
#X obj 376 77 t f f;
200 #X obj 379 97 *;
#X obj 345 260 *~ 0;
#X msg 0 238 COMPRESS \, compress 97 88 0.5 \, set 100 40 150;
#X obj -1 216 loadbang;
#X obj 345 173 sel 0 3 7 12;
205 #X obj 347 152 mod 16;
#X obj 440 231 snare~;
#X obj 439 208 f 64;
#X obj 439 259 *~ 0;
#X obj 430 59 / 127;

```

```

210 #X obj 438 114 * 8;
    #X obj 436 77 t f f;
    #X obj 439 97 *;
    #X obj 455 137 div 2;
    #X obj 439 162 mod 16;
215 #X obj 439 183 sel 4 10;
    #X obj 289 211 sel 4 10;
    #X obj 526 233 hihat~;
    #X obj 529 162 mod 4;
    #X obj 545 137 div 3;
220 #X obj 526 259 *~ 1;
    #X obj 597 139 div 3;
    #X obj 671 251 *~ 1;
    #X obj 529 183 sel 1 2 3;
    #X obj 623 185 + 48;
225 #X obj 529 208 f 0;
    #X obj 558 209 f 48;
    #X obj 599 157 moses 4;
    #X msg 595 184 0;
    #X obj 166 88 mod 64;
230 #X obj 179 115 / 63;
    #X obj 672 228 clap~;
    #X obj 672 159 mod 16;
    #X obj 672 183 sel 7;
    #X obj 672 206 f 64;
235 #X obj 719 163 div 4;
    #X obj 718 187 * 2;
    #X obj 338 35 route 76 77 78 79 80 81 74 75;
    #X obj 204 219 *~;
    #X obj 134 219 *~;
240 #X obj 218 196 vline~;
    #X msg 218 177 \$1 10;
    #X obj 79 50 route 32 40 66 24;
    #X obj 185 73 / 127;
    #X obj 257 143 sel 0 6 12;
245 #X obj 247 97 mod 64;
    #X obj 248 122 mod 16;
    #X msg 289 235 2;
    #X obj 531 33 vline~;
    #X msg 531 12 0 \, 1 \$1;
250 #X msg 531 -31 2560;
    #X obj 531 -12 * 444;
    #X obj 277 0 delay;
    #X msg 276 21 0;
    #X msg 250 -12 1;
255 #X obj 362 -25 bng 15 250 50 0 empty empty empty 17 7 0 10 -262144
    -1 -1;
    #X obj 624 -27 snapshot~;
    #X obj 627 0 hsl 128 15 0 1 0 0 empty empty empty -2 -8 0 10 -262144
    -1 -1 10835 1;
260 #X obj 530 101 expr~ if($v1<$v2 \, $v1/$v2 \, (1-$v1)/(1-$v2));
    #X obj 535 428 *~;
    #X obj 531 81 wrap~;
    #X obj 535 337 wrap~;
    #X obj 530 59 *~ 320;
265 #X obj 537 377 expr~ if($v1<$v2 \, $v1/$v2 \, (1-$v1)/(1-$v2));
    #X obj 614 348 wrap~;

```

```

#X obj 614 445 expr~ if($v1<$v2 \, $v1/$v2 \, (1-$v1)/(1-$v2));
#X obj 655 481 hip~ 5;
#X obj 615 236 *~ 0.444;
270 #X obj 615 214 *~ 2560;
#X obj 615 323 *~ 4;
#X obj 654 353 wrap~;
#X obj 654 425 expr~ if($v1<$v2 \, $v1/$v2 \, (1-$v1)/(1-$v2));
#X obj 696 352 wrap~;
275 #X obj 694 405 expr~ if($v1<$v2 \, $v1/$v2 \, (1-$v1)/(1-$v2));
#X obj 653 329 *~ 5;
#X obj 694 331 *~ 6;
#X obj 536 402 *~;
#X obj 534 318 *~ 4;
280 #X obj 764 233 sel 0 2 3;
#X obj 764 211 change;
#X obj 614 255 *~ 10;
#X obj 615 293 *~ 5;
#X msg 765 255 5;
285 #X msg 785 270 6;
#X msg 794 288 7;
#X obj 765 192 div 32;
#X obj 765 171 mod 128;
#X obj 535 292 *~;
290 #X obj 247 69 f 0;
#X obj 767 26 div 128;
#X obj 768 49 change;
#X floatatom 767 69 5 0 0 0 - - -;
#X connect 2 0 5 1;
295 #X connect 2 0 6 0;
#X connect 3 0 6 1;
#X connect 4 0 5 0;
#X connect 5 0 0 0;
#X connect 6 0 1 0;
300 #X connect 7 0 4 0;
#X connect 7 0 2 0;
#X connect 8 0 3 0;
#X connect 8 0 2 1;
#X connect 9 0 74 0;
305 #X connect 9 1 73 0;
#X connect 10 0 77 0;
#X connect 11 0 16 0;
#X connect 12 0 15 0;
#X connect 13 0 14 0;
310 #X connect 14 0 9 0;
#X connect 15 0 9 0;
#X connect 16 0 9 0;
#X connect 17 0 7 0;
#X connect 18 0 8 0;
315 #X connect 19 0 18 1;
#X connect 19 0 17 1;
#X connect 20 0 91 0;
#X connect 20 0 123 0;
#X connect 21 0 20 0;
320 #X connect 22 0 123 1;
#X connect 23 0 19 0;
#X connect 24 0 19 0;
#X connect 25 0 19 0;

```

```

#X connect 26 0 36 0;
325 #X connect 27 0 26 0;
#X connect 28 0 30 0;
#X connect 29 0 27 1;
#X connect 30 0 31 0;
#X connect 31 0 29 0;
330 #X connect 32 0 34 0;
#X connect 33 0 36 1;
#X connect 34 0 35 0;
#X connect 34 1 35 1;
#X connect 35 0 33 0;
335 #X connect 36 0 8 0;
#X connect 36 0 7 0;
#X connect 37 0 2 0;
#X connect 38 0 37 0;
#X connect 39 0 27 0;
340 #X connect 39 1 27 0;
#X connect 39 2 27 0;
#X connect 39 3 27 0;
#X connect 40 0 39 0;
#X connect 41 0 43 0;
345 #X connect 42 0 41 0;
#X connect 43 0 8 0;
#X connect 43 0 7 0;
#X connect 44 0 46 0;
#X connect 45 0 43 1;
350 #X connect 46 0 47 0;
#X connect 46 1 47 1;
#X connect 47 0 45 0;
#X connect 48 0 42 1;
#X connect 49 0 50 0;
355 #X connect 50 0 42 0;
#X connect 50 1 42 0;
#X connect 51 0 82 0;
#X connect 51 1 82 0;
#X connect 52 0 55 0;
360 #X connect 53 0 58 0;
#X connect 54 0 60 1;
#X connect 55 0 8 0;
#X connect 55 0 7 0;
#X connect 56 0 62 0;
365 #X connect 57 0 8 0;
#X connect 57 0 7 0;
#X connect 58 0 60 0;
#X connect 58 1 61 0;
#X connect 58 2 60 0;
370 #X connect 59 0 61 1;
#X connect 60 0 52 0;
#X connect 61 0 52 0;
#X connect 62 0 63 0;
#X connect 62 1 59 0;
375 #X connect 63 0 61 0;
#X connect 64 0 65 0;
#X connect 65 0 13 0;
#X connect 66 0 57 0;
#X connect 67 0 68 0;
380 #X connect 68 0 69 0;

```

```

#X connect 69 0 66 0;
#X connect 70 0 71 0;
#X connect 71 0 69 1;
#X connect 72 0 28 0;
385 #X connect 72 1 32 0;
#X connect 72 2 48 0;
#X connect 72 3 44 0;
#X connect 72 4 54 0;
#X connect 72 5 56 0;
390 #X connect 72 6 70 0;
#X connect 73 0 18 0;
#X connect 74 0 17 0;
#X connect 75 0 73 1;
#X connect 75 0 74 1;
395 #X connect 76 0 75 0;
#X connect 77 0 11 0;
#X connect 77 1 12 0;
#X connect 77 2 64 0;
#X connect 77 3 78 0;
400 #X connect 77 4 72 0;
#X connect 78 0 76 0;
#X connect 79 0 23 0;
#X connect 79 1 24 0;
#X connect 79 2 25 0;
405 #X connect 79 3 51 0;
#X connect 80 0 81 0;
#X connect 81 0 79 0;
#X connect 82 0 19 0;
#X connect 83 0 91 0;
410 #X connect 83 0 93 1;
#X connect 83 0 97 0;
#X connect 83 0 100 1;
#X connect 83 0 103 0;
#X connect 83 0 106 1;
415 #X connect 83 0 108 1;
#X connect 84 0 83 0;
#X connect 85 0 86 0;
#X connect 86 0 84 0;
#X connect 86 0 87 0;
420 #X connect 87 0 88 0;
#X connect 88 0 20 0;
#X connect 88 0 123 1;
#X connect 89 0 20 0;
#X connect 90 0 89 0;
425 #X connect 90 0 85 0;
#X connect 91 0 92 0;
#X connect 93 0 122 0;
#X connect 93 0 122 1;
#X connect 94 0 8 0;
430 #X connect 94 0 7 0;
#X connect 95 0 93 0;
#X connect 95 0 98 1;
#X connect 96 0 98 0;
#X connect 97 0 95 0;
435 #X connect 98 0 111 0;
#X connect 98 0 111 1;
#X connect 99 0 100 0;

```

```

#X connect 100 0 101 0;
#X connect 101 0 94 1;
440 #X connect 102 0 115 0;
#X connect 103 0 102 0;
#X connect 104 0 99 0;
#X connect 105 0 106 0;
#X connect 106 0 101 0;
445 #X connect 107 0 108 0;
#X connect 108 0 101 0;
#X connect 109 0 105 0;
#X connect 110 0 107 0;
#X connect 111 0 94 0;
450 #X connect 112 0 96 0;
#X connect 113 0 117 0;
#X connect 113 1 118 0;
#X connect 113 2 119 0;
#X connect 114 0 113 0;
455 #X connect 115 0 116 0;
#X connect 116 0 104 0;
#X connect 116 0 109 0;
#X connect 116 0 110 0;
#X connect 117 0 116 1;
460 #X connect 118 0 116 1;
#X connect 119 0 116 1;
#X connect 120 0 114 0;
#X connect 121 0 120 0;
#X connect 122 0 112 0;
465 #X connect 123 0 22 0;
#X connect 123 0 40 0;
#X connect 123 0 49 0;
#X connect 123 0 53 0;
#X connect 123 0 67 0;
470 #X connect 123 0 80 0;
#X connect 123 0 121 0;
#X connect 123 0 124 0;
#X connect 124 0 125 0;
#X connect 125 0 126 0;
475 #X restore 21 406 pd \ $0-audio;
#X connect 1 0 6 0;
#X connect 1 0 10 1;
#X connect 2 0 1 0;
#X connect 3 0 4 0;
480 #X connect 4 0 10 0;
#X connect 5 0 2 0;
#X connect 6 0 5 1;
#X connect 7 0 9 0;
#X connect 8 0 9 0;
485 #X connect 9 0 5 0;
#X connect 10 0 1 0;
#X connect 11 0 12 0;
#X connect 12 0 13 0;
#X connect 14 0 15 0;
490 #X connect 14 1 15 1;
#X connect 14 2 15 2;
#X connect 15 0 16 0;
#X connect 15 0 25 0;
#X connect 16 0 17 0;

```

```

495  #X connect 18 0 19 0;
    #X connect 18 0 114 0;
    #X connect 18 0 156 0;
    #X connect 19 0 21 0;
    #X connect 19 0 82 0;
500  #X connect 19 1 27 0;
    #X connect 19 1 85 0;
    #X connect 19 2 30 0;
    #X connect 19 2 88 0;
    #X connect 19 3 33 0;
505  #X connect 19 3 91 0;
    #X connect 19 4 36 0;
    #X connect 19 4 94 0;
    #X connect 19 5 39 0;
    #X connect 19 5 97 0;
510  #X connect 19 8 53 0;
    #X connect 20 0 22 0;
    #X connect 20 1 23 0;
    #X connect 21 0 20 0;
    #X connect 22 0 24 0;
515  #X connect 23 0 24 0;
    #X connect 24 0 51 0;
    #X connect 25 0 18 0;
    #X connect 26 0 41 0;
    #X connect 26 1 42 0;
520  #X connect 27 0 26 0;
    #X connect 28 0 51 0;
    #X connect 29 0 43 0;
    #X connect 29 1 44 0;
    #X connect 30 0 29 0;
525  #X connect 31 0 51 0;
    #X connect 32 0 45 0;
    #X connect 32 1 46 0;
    #X connect 33 0 32 0;
    #X connect 34 0 51 0;
530  #X connect 35 0 47 0;
    #X connect 35 1 48 0;
    #X connect 36 0 35 0;
    #X connect 37 0 51 0;
    #X connect 38 0 49 0;
535  #X connect 38 1 50 0;
    #X connect 39 0 38 0;
    #X connect 40 0 51 0;
    #X connect 41 0 28 0;
    #X connect 42 0 28 0;
540  #X connect 43 0 31 0;
    #X connect 44 0 31 0;
    #X connect 45 0 34 0;
    #X connect 46 0 34 0;
    #X connect 47 0 37 0;
545  #X connect 48 0 37 0;
    #X connect 49 0 40 0;
    #X connect 50 0 40 0;
    #X connect 51 0 3 0;
    #X connect 53 0 55 0;
550  #X connect 53 1 54 0;
    #X connect 53 2 56 0;

```

```
#X connect 53 3 57 0;
#X connect 53 4 63 0;
#X connect 54 0 55 0;
555 #X connect 54 1 55 1;
#X connect 55 0 58 0;
#X connect 56 0 55 0;
#X connect 56 1 55 2;
#X connect 57 0 55 0;
560 #X connect 57 1 55 3;
#X connect 58 0 51 0;
#X connect 59 0 60 0;
#X connect 59 1 60 1;
#X connect 60 0 64 0;
565 #X connect 61 0 60 0;
#X connect 61 1 60 2;
#X connect 62 0 60 0;
#X connect 62 1 60 3;
#X connect 63 0 70 0;
570 #X connect 63 1 71 0;
#X connect 63 2 72 0;
#X connect 63 3 62 0;
#X connect 63 4 133 0;
#X connect 64 0 65 0;
575 #X connect 64 1 66 0;
#X connect 64 2 67 0;
#X connect 64 3 67 1;
#X connect 64 3 66 1;
#X connect 64 3 65 1;
580 #X connect 65 0 73 0;
#X connect 66 0 74 0;
#X connect 67 0 75 0;
#X connect 68 0 69 0;
#X connect 69 0 51 0;
585 #X connect 70 0 60 0;
#X connect 71 0 59 0;
#X connect 72 0 61 0;
#X connect 73 0 68 0;
#X connect 74 0 68 1;
590 #X connect 75 0 68 2;
#X connect 82 0 83 0;
#X connect 82 1 84 0;
#X connect 83 0 81 0;
#X connect 84 0 100 0;
595 #X connect 85 0 86 0;
#X connect 85 1 87 0;
#X connect 86 0 77 0;
#X connect 87 0 101 0;
#X connect 88 0 89 0;
600 #X connect 88 1 90 0;
#X connect 89 0 76 0;
#X connect 90 0 102 0;
#X connect 91 0 92 0;
#X connect 91 1 93 0;
605 #X connect 92 0 78 0;
#X connect 93 0 103 0;
#X connect 94 0 95 0;
#X connect 94 1 96 0;
```

```

#X connect 95 0 79 0;
610 #X connect 96 0 104 0;
#X connect 97 0 98 0;
#X connect 97 1 99 0;
#X connect 98 0 80 0;
#X connect 99 0 105 0;
615 #X connect 100 0 81 0;
#X connect 101 0 77 0;
#X connect 102 0 76 0;
#X connect 103 0 78 0;
#X connect 104 0 79 0;
620 #X connect 105 0 80 0;
#X connect 106 0 107 0;
#X connect 106 0 137 0;
#X connect 106 0 140 0;
#X connect 107 0 110 0;
625 #X connect 108 0 109 0;
#X connect 109 0 106 0;
#X connect 110 0 106 1;
#X connect 112 0 155 0;
#X connect 112 0 126 0;
630 #X connect 115 0 117 0;
#X connect 116 0 115 0;
#X connect 117 0 118 0;
#X connect 117 0 121 0;
#X connect 118 0 119 0;
635 #X connect 119 0 117 1;
#X connect 120 0 125 0;
#X connect 121 0 120 0;
#X connect 121 1 122 0;
#X connect 122 0 125 1;
640 #X connect 123 0 122 1;
#X connect 126 0 127 0;
#X connect 128 0 116 0;
#X connect 128 1 123 0;
#X connect 129 0 130 0;
645 #X connect 130 0 131 0;
#X connect 130 0 132 1;
#X connect 132 0 128 0;
#X connect 133 0 134 0;
#X connect 133 1 129 0;
650 #X connect 134 0 135 0;
#X connect 135 0 136 0;
#X connect 136 0 132 0;
#X connect 138 0 142 0;
#X connect 139 0 138 0;
655 #X connect 140 0 139 0;
#X connect 140 1 141 0;
#X connect 141 0 149 1;
#X connect 142 0 143 0;
#X connect 142 0 145 0;
660 #X connect 143 0 144 0;
#X connect 144 0 142 1;
#X connect 145 0 147 0;
#X connect 145 1 146 0;
#X connect 146 0 147 1;
665 #X connect 147 0 148 0;
```

```

#X connect 148 0 149 0;
#X connect 148 1 152 1;
#X connect 149 0 150 0;
#X connect 150 0 151 0;
670 #X connect 151 1 152 0;
#X connect 152 0 153 0;
#X connect 152 1 153 1;
#X connect 153 0 19 0;
#X connect 155 0 111 0;
675 #X connect 156 0 52 0;
#X connect 156 1 52 1;

```

## 20 -/reflex3d2d.hs

```

module Main where

import Control.Monad(forever, when)
import Data.IORef(IORef, newIORef, atomicModifyIORef)
5 import Control.Concurrent(forkIO, yield)
import System.IO(hGetBuf, withBinaryFile, IOMode(ReadMode), hSetBuffering, ↵
    ↵ BufferMode(LineBuffering), stdin)
import Foreign.Marshal.Alloc(allocaBytes)
import Foreign.Marshal.Array(peekArray)
import Foreign.Storable(sizeOf)
10 import Graphics.UI.GLUT hiding(Point, project)

import Vector hiding(i, o)

fr3 :: Int -> Int -> [V3]
15 fr3 p q =
    let v0 = V3 1 0 0
        v1 = m1 ^^* v0
        v2 = (m1 ^^* m2) ^^* v0
        m1 = M3 c1 s1 0 (-s1) c1 0 0 0 1
        20 m2 = M3 c2 0 s2 0 1 0 (-s2) 0 c2
        c1 = r1 / r0
        c2 = r2 / r1
        s1 = sqrt (1 - c1*c1)
        s2 = sqrt (1 - c2*c2)
        25 r0 = radius 0 [p,q]
        r1 = radius 1 [p,q]
        r2 = radius 2 [p,q]
    in [ norm $ cross3 v1 v2
        , norm $ cross3 v0 v2 ^^* (-1)
        30 , norm $ cross3 v0 v1
        , v0, v1, v2 ]

delta :: [Int] -> R
delta [] = 1
35 delta [p] = let s = sin (pi / fromIntegral p) in s * s
delta (p:q:rs) = let c = cos (pi / fromIntegral p) in delta (q:rs) - delta rs * ↵
    ↵ c * c

radius2 :: Int -> [Int] -> R
radius2 0 [] = 1
40 radius2 0 (p:qs) = delta qs / delta (p:qs)
radius2 j ps = radius2 0 ps - radius2 0 (take (j-1) ps)

```

```

radius :: Int -> [Int] -> R
radius n ps = sqrt $ radius2 n ps

45 dropV :: V v => v -> v -> v
dropV u v =
    let k = u `dot` v / u `dot` u
    in v ^-^ (k *^ u)

50 data Figure3 = Figure3 !V3 !V3 !V3 !V3 !V3 !V3 !V3 !Bool

instance Approx Figure3 where
    Figure3 p a b c _ _ _ _ ~~= Figure3 q e f g _ _ _ _ = and $ zipWith (~=) [p,a,
        ↪ ,b,c] [q,e,f,g]

55 initialP3 :: [V3] -> V3 -> V3
initialP3 [V3 a11 a12 a13, V3 a21 a22 a23, V3 a31 a32 a33, _, _, _] v =
    let m = M3 a11 a12 a13 a21 a22 a23 a31 a32 a33
        u = norm $ inv m ^^^ v
60     in u
initialP3 _ _ = error "initialP3"

initialF3 :: [V3] -> V3 -> Figure3
initialF3 [l0,l1,l2,v0,_,v2] p = Figure3 p a b c d e f True
65     where (a, b, c) = (dropV l0 p , dropV l1 p , dropV l2 p)
        (d, e, f) = (v2 , v0 , (dropV l0 . dropV l2) p)
initialF3 _ _ = error "initialF3"

transformF3 :: M3 -> Figure3 -> Figure3
70 transformF3 m (Figure3 p a b c d e f z) = Figure3 (m ^^^ p) (m ^^^ a) (m ^^^ ↪
    ↪ b) (m ^^^ c) (m ^^^ d) (m ^^^ e) (m ^^^ f) (if det m > 0 then z else ↪
    ↪ not z)

transformsF3 :: M3 -> [Figure3] -> [Figure3]
transformsF3 m fs = map (m `transformF3`) fs

75 polyhedronF :: [M3] -> Figure3 -> [Figure3]
polyhedronF ms f = map ('transformF3' f) ms

quality = 6

80 drawFigure3 :: (Color3 R, Color3 R, Color3 R) -> (Bool, Bool, Bool) -> Figure3 ↪
    ↪ -> IO ()
drawFigure3 (ac,bc,cc) (av,bv,cv) (Figure3 p a b c d e f z) = do
    let seg p1 p2 = let V2 x1 y1 = project p1
        V2 x2 y2 = project p2
85         in vertex (Vertex2 x1 y1) >> vertex (Vertex2 x2 y2)
    lin p1 p2 n = let dir = (p2 ^-^ p1 ^/ fromIntegral n) in map (\i -> norm $ ↪
        ↪ p1 ^+^ (fromIntegral i *^ dir)) [0 .. n :: Int]
    line p1 p2 = let ps = lin p1 p2 quality in sequence_ $ zipWith seg ps (↪
        ↪ tail ps)
    when av $ color ac >> line p a >> line p d
    when bv $ color bc >> line p b >> line p e
90    when cv $ color cc >> line p c >> line p f

drawFigures3 :: (Color3 R, Color3 R, Color3 R) -> (Bool, Bool, Bool) -> [Figure3 ↪

```

```

    ↪ ] -> IO ()
drawFigures3 cs v fs = renderPrimitive Lines $ mapM_ (drawFigure3 cs v) fs

95
project :: V3 -> V2
project p = let (V3 x y z) = norm p in let k = 0.5 / (1 - z) in V2 (k*x) (k*y)

fillFigure3 :: (Color3 R, Color3 R, Color3 R) -> (Bool, Bool, Bool) -> Figure3 ↪
    ↪ -> IO ()
100 fillFigure3 (ac,bc,cc) (av,bv,cv) (Figure3 p a b c d e f parity) = do
    when av $ color ac >> triangle' p a d >> triangle' p d b
    when bv $ color bc >> triangle' p b e >> triangle' p e c
    when cv $ color cc >> triangle' p c f >> triangle' p f a
    where
105     triangle' p1 p2 p3 = triangle parity quality p1 p2 p3

clockwise :: V2 -> V2 -> V2 -> Bool
clockwise (V2 ax ay) (V2 bx by) (V2 cx cy) = elx * e2y - ely * e2x >= 0
    where
110     (elx, ely) = (ax - bx, ay - by)
     (e2x, e2y) = (cx - bx, cy - by)

triangle :: Bool -> Int -> V3 -> V3 -> V3 -> IO ()
triangle cw 0 p1 p2 p3 = do
115     let v1@(V2 x1 y1) = project p1
         v2@(V2 x2 y2) = project p2
         v3@(V2 x3 y3) = project p3
    when (clockwise v1 v2 v3 == cw) $ do
        vertex (Vertex2 x1 y1)
120     vertex (Vertex2 x2 y2)
        vertex (Vertex2 x3 y3)

triangle cw n p1 p2 p3 = do
    let q1 = 0.5 *^ (p2 ^+^ p3)
125     q2 = 0.5 *^ (p3 ^+^ p1)
        q3 = 0.5 *^ (p1 ^+^ p2)
    triangle (not cw) (n-1) p1 q2 q3
    triangle (not cw) (n-1) q1 p2 q3
    triangle (not cw) (n-1) q1 q2 p3
130     triangle cw (n-1) q1 q2 q3

fillFigures3 :: (Color3 R, Color3 R, Color3 R) -> (Bool, Bool, Bool) -> [Figure3 ↪
    ↪ ] -> IO ()
fillFigures3 cs v fs = renderPrimitive Triangles $ mapM_ (fillFigure3 cs v) fs

135 data State = State { sV :: Size, sVis :: (Bool, Bool, Bool), sP0 :: Point, sP :: ↪
    ↪ Polytope, sFrame :: Int }

reshape :: IORef State -> Size -> IO ()
reshape r v = atomicModifyIORef r $ \s -> (s { sV = v }, ())

140 timer :: Timeout -> IORef State -> IO ()
timer t r = do
    addTimerCallback t $ timer t r
    postRedisplay Nothing

145 data Polytope = Polytope [M3] [V3]

```

```

data Point = Point V3

red :: Color3 R
red = Color3 ((0.114/0.299)**0.2) 0 0
150
green :: Color3 R
green = Color3 0 ((0.114/0.587)**0.2) 0

blue :: Color3 R
155 blue = Color3 0 0 1

display :: IORef State -> IO ()
display ref = do
  s <- atomicModifyIORef ref $ \x -> (x, x)
160  let vp@(Size vw vh) = sV s
      w = 2 * fromIntegral vw / fromIntegral vh
      h = 2
      viewport $= (Position 0 0, vp)
      matrixMode $= Projection
165  loadIdentity
      ortho (-w) (w) (-h) (h) (-1) 1
      matrixMode $= Modelview 0
      loadIdentity
      clear [ColorBuffer, DepthBuffer]
170  depthFunc $= Just Less
      shadeModel $= Smooth
      lineSmooth $= Enabled
      lineWidth $= 2
      let Point p = sP0 s
175      Polytope syms fr = sP s
          a1 = fromIntegral (sFrame s) / 100
          a2 = fromIntegral (sFrame s) / 141
          a3 = fromIntegral (sFrame s) / 382
          c1 = cos a1
180          s1 = sin a1
          c2 = cos a2
          s2 = sin a2
          c3 = cos a3
          s3 = sin a3
185          r1 = M3 c1 s1 0 (-s1) c1 0 0 0 1
          r2 = M3 c2 0 s2 0 1 0 (-s2) 0 c2
          r3 = M3 1 0 0 0 c3 s3 0 (-s3) c3
          r = r1 ^^*^^ r2 ^^*^^ r3
      fillFigures3 (red, green, blue) (sVis s) . transformsF3 r . polyhedronF syms . ↵
          ↵ initialF3 fr . initialP3 fr $ p
190  flush
      swapBuffers
      atomicModifyIORef ref $ \s'' -> (s'' { sFrame = sFrame s'' + 1 }, ())

main :: IO ()
195 main = do
  hSetBuffering stdin LineBuffering
  let v = Size 1024 768
  initialWindowSize $= v
  initialDisplayMode $= [RGBAMode, DoubleBuffered, WithDepthBuffer]
200  (_, dims) <- getArgsAndInitialize
  let [p,q] = map read dims

```

```

Just (file , size) = lookup (p,q) [((3,3),("3_3.3d", 24))
                                   ,((4,3),("4_3.3d", 48))
                                   ,((5,3),("5_3.3d", 120))]
205     n = 3*3 * size
        b = n * sizeOf (undefined::R)
matrices <- fmap toMats $ withBinaryFile file ReadMode $ \h -> do
  allocaBytes b $ \ptr -> do
    hGetBuf h ptr b
210     peekArray n ptr
        createWindow "reflex"
        sr <- newIORef $ State { sV = v, sVis = (True, True, True), sP0 = Point (V3 1 1 1), sP = Polytope matrices (fr3 p q), sFrame = 500 }
        forkIO . forever $ commands sr
        reshapeCallback $= Just (reshape sr)
215     displayCallback $= display sr
        idleCallback $= Just yield
        timer 50 sr
        mainLoop

220 toMats :: [R] -> [M3]
toMats (a:b:c:d:e:f:g:h:i:zz) = M3 a b c d e f g h i : toMats zz
toMats _ = []

commands :: IORef State -> IO ()
225 commands r = do
  l <- getLine
  case map read (words l) of
    [la,lb,lc,va,vb,vc] -> atomicModifyIORef r $ \s -> (s { sVis = (va/=0,vb/=0,vc/=0), sP0 = Point (V3 la lb lc) }, ())
    _ -> return ()
230 postRedisplay Nothing

```

## 21 -/reflex-44100.sh

```

#!/bin/bash
pd -stderr -rt -nadc -alsa -audiooutdev 6 -audiobuf 100 -outchannels 2 \
-open REFLEX-nano.pd 2>&1 |
DISPLAY=":0.1" ./reflex

```

## 22 reflex.hs

```

module Main(main) where

import Control.Monad(liftM2, forever, when)
import Data.List(deleteFirstsBy, nubBy, maximumBy)
5 import Data.Ord (comparing)
import Data.IORef(IORef, newIORef, atomicModifyIORef)
import Control.Concurrent(forkIO)
import Control.Applicative
import System.IO(hGetBuf, withBinaryFile, IOMode(ReadMode), hSetBuffering, \
  ↪ BufferMode(LineBuffering), stdin, stdout)
10 import Foreign.Marshal.Alloc(allocaBytes)
import Foreign.Marshal.Array(peekArray, withArray)
import Foreign.Ptr (Ptr, castPtr)
import Foreign.Storable(sizeOf)
import Graphics.UI.GLUT

```

```

15  import Snapshot
    import Vector hiding (i, o)
    import qualified Vector as V
    import Ball

20  fr4 :: Int -> Int -> Int -> [V4]
    fr4 p q r =
        let v0 = V4 1 0 0 0
            v1 = m1 ^^*^ v0
25      v2 = (m1 ^^*^ m2) ^^*^ v0
            v3 = ((m1 ^^*^ m2) ^^*^ m3) ^^*^ v0
            m1 = M4 c1 s1 0 0 (-s1) c1 0 0 0 0 1 0 0 0 0 1
            m2 = M4 c2 0 s2 0 0 1 0 0 (-s2) 0 c2 0 0 0 0 1
            m3 = M4 c3 0 0 s3 0 1 0 0 0 0 1 0 (-s3) 0 0 c3
30      c1 = r1 / r0
            c2 = r2 / r1
            c3 = r3 / r2
            s1 = sqrt (1 - c1*c1)
            s2 = sqrt (1 - c2*c2)
35      s3 = sqrt (1 - c3*c3)
            r0 = radius 0 [p,q,r]
            r1 = radius 1 [p,q,r]
            r2 = radius 2 [p,q,r]
            r3 = radius 3 [p,q,r]
40      l0 = norm $ cross4 v1 v2 v3
            l1 = norm $ cross4 v0 v2 v3 ^* (-1)
            l2 = norm $ cross4 v0 v1 v3
            l3 = norm $ cross4 v0 v1 v2 ^* (-1)
        in [l0, l1, l2, l3]

45  delta :: [Int] -> R
    delta [] = 1
    delta [p] = let s = sin (pi / fromIntegral p) in s * s
    delta (p:q:rs) = let c = cos (pi / fromIntegral p) in delta (q:rs) - delta rs * ↵
        ↵ c * c

50  radius2 :: Int -> [Int] -> R
    radius2 0 [] = 1
    radius2 0 (p:qs) = delta qs / delta (p:qs)
    radius2 j ps = radius2 0 ps - radius2 0 (take (j-1) ps)

55  radius :: Int -> [Int] -> R
    radius n ps = sqrt $ radius2 n ps

    data Space4d1 a = Space4d1 { x, y, z, w :: a }
60  instance Functor Space4d1 where
        fmap f (Space4d1 a b c d) = Space4d1 (f a) (f b) (f c) (f d)
    instance Applicative Space4d1 where
        pure f = Space4d1 f f f f
        Space4d1 a b c d <*> Space4d1 i j k l = Space4d1 (a i) (b j) (c k) (d l)

65  data Space4d2 a = Space4d2 { xy, xz, xw, yz, yw, zw :: a }
    instance Functor Space4d2 where
        fmap f (Space4d2 a b c d e g) = Space4d2 (f a) (f b) (f c) (f d) (f e) (f g)
    instance Applicative Space4d2 where
70      pure f = Space4d2 f f f f f f

```

```

Space4d2 a b c d e f <*> Space4d2 i j k l m n = Space4d2 (a i) (b j) (c k) (d l)
    (e m) (f n)

data Space4d3 a = Space4d3 { xyz, xyw, xzw, yzw :: a }
instance Functor Space4d3 where
75   fmap f (Space4d3 a b c d) = Space4d3 (f a) (f b) (f c) (f d)
instance Applicative Space4d3 where
    pure f = Space4d3 f f f f
    Space4d3 a b c d <*> Space4d3 i j k l = Space4d3 (a i) (b j) (c k) (d l)

80 data Figure4 a = Figure4 a (Space4d1 a) (Space4d2 a) (Space4d3 a)
instance Functor Figure4 where
    fmap f (Figure4 o ps qs rs) = Figure4 (f o) (fmap f ps) (fmap f qs) (fmap f rs)
instance Applicative Figure4 where
    pure f = Figure4 f (pure f) (pure f) (pure f)
85   Figure4 a bs cs ds <*> Figure4 i js ks ls = Figure4 (a i) (bs <*> js) (cs <*>
    (ds <*> ls))

type F4 = Figure4 V4

90 initialP4 :: [V4] -> V4 -> V4
initialP4 [V4 a11 a12 a13 a14, V4 a21 a22 a23 a24, V4 a31 a32 a33 a34, V4 a41
    (a42 a43 a44)] v =
    let m = M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44
        u = norm $ inv m ^^^ v
95   in u
initialP4 _ _ = error "initialP4"

initialF4 :: [V4] -> V4 -> F4
initialF4 [10,11,12,13] p = let munge = norm . center p . subgroup in Figure4
100   (norm p)
    (Space4d1
      (munge [10])
      (munge [11])
      (munge [12])
105   (munge [13]))
    (Space4d2
      (munge [10, 11])
      (munge [10, 12])
      (munge [10, 13])
110   (munge [11, 12])
      (munge [11, 13])
      (munge [12, 13]))
    (Space4d3
      (munge [10, 11, 12])
115   (munge [10, 11, 13])
      (munge [10, 12, 13])
      (munge [11, 12, 13]))
    initialF4 _ _ = error "initialF4"

120 closure :: (Approx m, M m) => [m] -> [m]
closure gs = closure' gs gs gs

closure' :: (Approx m, M m) => [m] -> [m] -> [m] -> [m]

```

```

closure' gs ms ns =
125   let new = (deleteFirstsBy (~=) . nubBy (~=) . liftM2 (^*^^) gs $ ns) ms
       in if null new then ms else closure' gs (ms ++ new) new

subgroup :: [V4] -> [M4]
subgroup = closure . map reflector
130
center :: V4 -> [M4] -> V4
center v ms = (foldr (^+^) (V4 0 0 0 0) $ map (^*^^ v) ms) ^/ (fromIntegral $ \
    ↪ length ms)

transform :: M4 -> F4 -> F4
135 transform m = fmap (m ^*^^)

{-
transforms :: M4 -> [F4] -> [F4]
transforms m fs = map (m 'transform') fs
140 -}

polychoron :: [M4] -> F4 -> [F4]
polychoron ms f = map ('transform' f) ms

145 fillFigure4 :: R -> Space4d2 (Color3 R) -> Space4d2 Bool -> (Space4d3 V4 -> V4) ↪
    ↪ -> F4 -> F4 -> IO ()
fillFigure4 q cs vs - - {- axis (Figure4 - - - r0s) -} (Figure4 o ps qs -) = do
    let h = id -- if axis r0s ~=~ axis rs then fmap (1-) else id
        when (xy vs) $ face q (h $ xy cs) o (x ps) (y ps) (xy qs)
        when (xz vs) $ face q (h $ xz cs) o (x ps) (z ps) (xz qs)
150   when (xw vs) $ face q (h $ xw cs) o (x ps) (w ps) (xw qs)
        when (yz vs) $ face q (h $ yz cs) o (y ps) (z ps) (yz qs)
        when (yw vs) $ face q (h $ yw cs) o (y ps) (w ps) (yw qs)
        when (zw vs) $ face q (h $ zw cs) o (z ps) (w ps) (zw qs)

155 {-
project :: V4 -> Maybe V3
project (V4 xx yy zz ww) =
    let k = 1 / (1 - ww)
        v3 = V3 xx yy zz ^* k
160   in if ww < 0.99 then Just v3 else Nothing
-}

face :: R -> Color3 R -> V4 -> V4 -> V4 -> V4 -> IO ()
face q c p1 p2 p3 p4 = triangle q c (p1, p2, p3) >> triangle q c (p4, p2, p3)
165
{-
triangle' :: Color3 R -> (V4, V4, V4) -> IO ()
triangle' c (p1, p2, p3) =
    case liftM3 (,,) (project p1) (project p2) (project p3) of
170   Nothing -> return ()
       Just (v1, v2, v3) -> {- print (v1,v2,v3) >> -} draw v1 >> draw v2 >> draw v3
    where
        draw (V3 xx yy zz) = color (fmap (zz *) c) >> vertex (Vertex3 xx yy zz)
-}

175 triangle' :: Color3 R -> (V4, V4, V4) -> IO ()
triangle' c (p1, p2, p3) = draw p1 >> draw p2 >> draw p3
    where

```

```

draw (V4 xx yy zz ww) = color c >> vertex (Vertex4 xx yy zz ww)
180 triangle :: R -> Color3 R -> (V4, V4, V4) -> IO ()
triangle q c t0@(p1, p2, p3) = do
  let e1 = p2 | - | p3
      e2 = p3 | - | p1
185      e3 = p1 | - | p2
      q1 = norm $ p2 ^+^ p3
      q2 = norm $ p3 ^+^ p1
      q3 = norm $ p1 ^+^ p2
      (e, (t1, t2)) = maximumBy (comparing fst)
190      [ (e1, ((p1, p2, q1), (p1, q1, p3)))
        , (e2, ((p1, p2, q2), (q2, p2, p3)))
        , (e3, ((p1, q3, p3), (q3, p2, p3)))
        ]
  if e < q then triangle' c t0 else triangle q c t1 >> triangle q c t2
195 fillFigures4 :: R -> Space4d2 (Color3 R) -> Space4d2 Bool -> [F4] -> IO ()
fillFigures4 - - - [] = return ()
fillFigures4 q cs v fs@(f:_) = renderPrimitive Triangles $ mapM (fillFigure4 q ↗
  ↘ cs v xyz f) fs
200 data State = State { sView :: Size, sColour :: Space4d2 (Color3 R), sVisibility ↗
  ↘ :: Space4d2 Bool, sQuality :: R, sShape :: V4, sPolytope :: Polytope, ↗
  ↘ sTransform :: M4, sMove :: V3, sFrame :: Int, sProgram :: Program }

reshape :: IOREf State -> Size -> IO ()
reshape r v = atomicModifyIORef r $ \s -> (s { sView = v }, ())

205 timer :: Timeout -> IOREf State -> IO ()
timer t r = do
  addTimerCallback t $ timer t r
  postRedisplay Nothing

210 data Polytope = Polytope [M4] [V4]

red :: Color3 R
red = Color3 ((0.114/0.299)**0.2) 0 0

215 yellow :: Color3 R
yellow = Color3 ((0.114/0.299)**0.2) ((0.114/0.587)**0.2) 0

green :: Color3 R
green = Color3 0 ((0.114/0.587)**0.2) 0
220 cyan :: Color3 R
cyan = Color3 0 ((0.114/0.587)**0.2) 1

blue :: Color3 R
225 blue = Color3 0 0 1

magenta :: Color3 R
magenta = Color3 ((0.114/0.299)**0.2) 0 1

230 display :: IOREf State -> IO ()
display ref = do
  s <- atomicModifyIORef ref $ \s' -> (s', s')
```

```

    let vp@(Size vw vh) = sView s
        Polytope syms fr = sPolytope s
235     period = 25 * 30
        t = fromIntegral (sFrame s) / period
        shape = V4 (4 ** cos (2 * pi * 10 * t)) 1 1 1
--     mb = ball 64 $ sMove s
        co = cos (2 * pi / period)
240     si = sin (2 * pi / period)
        mb = M4 1 0 0 0 0 1 0 0 0 0 co si 0 0 (-si) co
        m = mb ^^*^^ sTransform s
    viewport $= (Position 0 0, vp)
    matrixMode $= Projection
245    loadIdentity
    perspective 45 (fromIntegral vw / fromIntegral vh) 1 20
    matrixMode $= Modelview 0
    loadIdentity
    multMatrix <=< (newMatrix RowMajor (matToList m) :: IO (GLmatrix R))
250    clear [ColorBuffer, DepthBuffer]
    depthFunc $= Just Less
    currentProgram $= Just (sProgram s)
    uloc <- get $ uniformLocation (sProgram s) "lights"
    withArray (concat [[xx,yy,zz, -0.5::GLfloat]|xx<-[-0.5,0,0.5],yy<-[-0.5,0,0.5],↵
        ↵ zz<-[-0.5,0,0.5], even . length . filter (== 0) $ [xx,yy,zz]]) $ \p -> ↵
        ↵ uniformv uloc 14 (castPtr p :: Ptr (Vertex4 GLfloat))
255    let
        fillFigures4 (sQuality s) (sColour s) (sVisibility s) . polychoron syms . ↵
            ↵ initialF4 fr . initialP4 fr $ shape
        currentProgram $= Nothing
        flush
        swapBuffers
260    reportErrors
    atomicModifyIORef ref $ \s' -> (s' { sFrame = sFrame s' + 1, sTransform = m },↵
        ↵ ())
--    when (t < 1) $ hSnapshot stdout (Position 0 0) vp

main :: IO ()
265 main = do
    hSetBuffering stdin LineBuffering
    let vp = Size 1050 576
    initialWindowSize $= vp
    initialDisplayMode $= [RGBAMode, DoubleBuffered, WithDepthBuffer]
270    (_, dims) <- getArgsAndInitialize
    let [p,q,r] = map read dims
        Just (file, size) = lookup (p,q,r) [((3,3,3),("3_3_3.4d", 120))
            ,((4,3,3),("4_3_3.4d", 384))
            ,((3,4,3),("3_4_3.4d", 1152))
275            ,((5,3,3),("5_3_3.4d", 14400))]

        n = 4*4 * size
        b = n * sizeof (undefined::R)
    matrices <- fmap toMats $ withBinaryFile file ReadMode $ \h -> do
        allocaBytes b $ \ptr -> do
280         _ <- hGetBuf h ptr b
        peekArray n ptr
        _ <- createWindow "reflex"
        prog <- shader vert frag
        let f = fr4 p q r
285    sr <- newIORef $ State

```

```

    { sView = vp
    , sColour = Space4d2 yellow yellow yellow yellow yellow yellow -- magenta ↯
      ↵ blue red cyan yellow green
    , sVisibility = Space4d2 True False True False True False
    , sQuality = 0.1
290   , sShape = V4 1 1 1 1
    , sPolytope = Polytope matrices f
    , sTransform = V.i -- reflector (norm $ ((f !! 0) ^-^ (f !! 1) ^+^ (f !! 2) ↯
      ↵ ^-^ (f !! 3)))
    , sMove = V3 0 0 0
    , sFrame = 0
295   , sProgram = prog
    }
_ <- forkIO . forever $ commands sr
reshapeCallback $= Just (reshape sr)
displayCallback $= display sr
300   timer 40 sr
      mainLoop

shader :: String -> String -> IO Program
shader v f = do
305   prog <- createProgram
      vv <- createShader VertexShader
      ff <- createShader FragmentShader
      shaderSource vv $= [v]
      shaderSource ff $= [f]
310   compileShader vv
      compileShader ff
      attachedShaders prog $= [vv, ff]
      linkProgram prog
      info <- get (programInfoLog prog)
315   when (not (null info)) $ putStrLn info
      return prog

matToList :: M4 -> [R]
matToList (M4 a b c d e f g h i j k l m n o p) = a:b:c:d:e:f:g:h:i:j:k:l:m:n:o:p↯
      ↵ :[]
320   toMats :: [R] -> [M4]
      toMats (a:b:c:d:e:f:g:h:i:j:k:l:m:n:o:p:zz) = M4 a b c d e f g h i j k l m n o p↯
      ↵ : toMats zz
      toMats _ = []

data Toggle
325   = On
    | Off
      deriving (Read, Show)

isOn :: Toggle -> Bool
330   isOn On = True
      isOn _ = False

data Command
    = Shape R R R R
335   | XY Toggle
    | XZ Toggle
    | XW Toggle
    | YZ Toggle

```

```

340   | YW Toggle
   | ZW Toggle
   | Move R R R
   | Quality R
   deriving (Read, Show)

345 commands :: IORef State -> IO ()
commands r = do
  l <- getLine
  let l' = (if strip then {- unwords . drop 1 . words . -} filter (/=';') else ↯
    ↪ id) l
  case reads l' of
350   (cmd,""):_ -> {- print cmd >> -} case cmd of
    Shape a b c d -> atomicModifyIORef r $ \s -> (s { sShape = V4 a b c d }, ↯
      ↪ ())
    XY v -> atomicModifyIORef r $ \s -> (s { sVisibility = (sVisibility s) { ↯
      ↪ xy = isOn v } }, ())
    XZ v -> atomicModifyIORef r $ \s -> (s { sVisibility = (sVisibility s) { ↯
      ↪ xz = isOn v } }, ())
    XW v -> atomicModifyIORef r $ \s -> (s { sVisibility = (sVisibility s) { ↯
      ↪ xw = isOn v } }, ())
355   YZ v -> atomicModifyIORef r $ \s -> (s { sVisibility = (sVisibility s) { ↯
      ↪ yz = isOn v } }, ())
    YW v -> atomicModifyIORef r $ \s -> (s { sVisibility = (sVisibility s) { ↯
      ↪ yw = isOn v } }, ())
    ZW v -> atomicModifyIORef r $ \s -> (s { sVisibility = (sVisibility s) { ↯
      ↪ zw = isOn v } }, ())
    Move i j k -> atomicModifyIORef r $ \s -> (s { sMove = V3 i j k }, ())
    Quality q | q > 0 -> atomicModifyIORef r $ \s -> (s { sQuality = recip q ↯
      ↪ }, ())
360   _ -> return ()
    _ -> return ()
  where strip = True

vert :: String
365 vert = unlines
  [ "void main(void) {"
    , "  vec4 v = gl_ModelViewMatrix * gl_Vertex;"
    , "  vec3 u = v.xyz / (1.0 - v.w);"
    , "  u *= 2.0;"
370   , "  u -= vec3(0.0, 0.0, 16.0);"
    , "  gl_Position = gl_ProjectionMatrix * vec4(u, 1.0);"
    , "  gl_TexCoord[0] = v;"
    , "  gl_TexCoord[1] = gl_Vertex;"
    , "  gl_FrontColor = gl_Color;"
375   , "}"
  ]

frag :: String
frag = unlines
380   [ "uniform vec4 lights[14];"
    , "void main(void) {"
    , "  float shine = 128.0;"
    , "  const vec3 light = vec3(0.0, 0.0, 50.0);"
    , "  vec3 c0 = gl_Color.rgb;"
385   , "  vec4 j = normalize(gl_TexCoord[1]) * 16.0;"
    , "  vec4 k = j + vec4(0.5);"

```

```

, " j -= floor(j);"
, " k -= floor(k);"
, " if (length(k - vec4(0.5)) < 0.5) {"
390 , "     shine /= 2.0;"
, "     }"
, " if (length(j - vec4(0.5)) < 0.5) {"
, "     shine *= 2.0;"
, "     }"
395 , " vec3 p = gl_FragCoord.xyz;"
, " vec4 q0 = normalize(gl_TexCoord[0]);"
, " vec3 q = q0.xyz / (1.0 - q0.w);"
, " vec3 a = normalize(dFdx(q));"
, " vec3 b = normalize(dFdy(q));"
400 , " vec3 n = normalize(cross(a, b));"
, " if (n.z > 0.0) n = -n;"
, " vec3 c = vec3(0.0);"
, " for (int i = 0; i < 14; ++i) {"
, "     vec3 l = 16.0 * lights[i].xyz;"
405 , "     vec3 d = normalize(q - l);"
, "     float dn = dot(d, n);"
, "     if (dn > 0.0) {"
, "         vec3 h = reflect(-d, n);"
, "         float diffuse = max(dn, 0.0) / 12.0;"
410 , "         float specular = pow(max(dot(h, n), 0.0), shine);"
, "         c += diffuse * c0 + (c0 + vec3(15.0)) * (specular / 24.0);"
, "     }"
, " }"
, " gl_FragColor = vec4(c / pow(max(-q.z, 1.0), 4.0), 1.0);"
415 , " gl_FragDepth = p.z;"
, " }"
]
```

## 23 -/reflexivity.pd

```

#N canvas 230 371 836 324 10;
#X obj 315 86 rx-visibility;
#X obj 321 270 print reflex;
#X obj 455 89 rx-move;
5 #X obj 172 89 rx-shape;
#X obj 314 -49 rx-solid;
#X connect 0 0 1 0;
#X connect 2 0 1 0;
#X connect 3 0 1 0;
10 #X connect 4 0 0 0;
```

## 24 -/reflex-jack.sh

```

#!/bin/bash
pd -stderr -nrt -jack -channels 2 \
-open reflexivity.pd 2>&1 |
DISPLAY=":0.1" ./reflex 3 4 3
```

## 25 -/REFLEX-nano.pd

```

#N canvas 120 0 1012 695 10;
#X obj 124 52 ctlin;
```

```

#X obj 122 87 pack f f f;
#X msg 121 108 \$3 \$2 \$1;
5 #X obj 121 133 route 1;
#X floatatom 182 64 5 0 0 0 - - -;
#X obj 120 155 route 2 3 4 5 23 24 25 26;
#X obj 88 534 print reflex;
#X obj 343 142 f 0;
10 #X obj 418 120 + 1;
#X obj 508 155 table \$0-sequence 768;
#X obj 507 133 s \$0-sequence;
#X obj 343 170 * 8;
#X obj 542 261 tabread \$0-sequence;
15 #X obj 345 214 t f f f f f f f;
#X obj 361 240 + 1;
#X obj 391 240 + 2;
#X obj 421 240 + 3;
#X obj 451 240 + 4;
20 #X obj 481 240 + 5;
#X obj 512 240 + 6;
#X obj 542 240 + 7;
#X obj 512 281 tabread \$0-sequence;
#X obj 481 301 tabread \$0-sequence;
25 #X obj 451 321 tabread \$0-sequence;
#X obj 421 341 tabread \$0-sequence;
#X obj 391 361 tabread \$0-sequence;
#X obj 361 382 tabread \$0-sequence;
#X obj 344 404 tabread \$0-sequence;
30 #X obj 344 460 pack f f f f f f f f;
#X obj 342 431 t f f;
#X obj 345 519 spigot;
#X obj 368 490 != -1;
#X obj 279 155 route 44;
35 #X obj 279 179 >= 64;
#X obj 151 278 pack f f f f f f f f;
#X obj 150 305 s \$0-sequence;
#X msg 508 108 const -1;
#X obj 344 77 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 1
40 1;
#X obj 344 191 t f f;
#X obj 396 548 unpack f f f f f f f f;
#X obj 393 737 dac~;
#X obj 355 587 / 127;
45 #X obj 392 587 / 127;
#X obj 431 587 / 127;
#X obj 467 587 / 127;
#X obj 365 621 pack f f f f;
#X obj 537 609 << 1;
50 #X obj 507 587 != 0;
#X obj 537 587 != 0;
#X obj 567 587 != 0;
#X obj 597 587 != 0;
#X obj 568 611 << 2;
55 #X obj 602 612 << 3;
#X obj 569 634 +;
#X obj 539 637 +;
#X obj 508 638 +;
#X obj 365 654 list prepend 0;

```

```
60 #N canvas 0 0 450 300 \$0-audio 1;
    #X obj 32 16 inlet;
    #X obj 117 262 outlet~;
    #X obj 169 263 outlet~;
    #X obj 31 44 route 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15;
65 #N canvas 0 0 450 300 \$0-1111 0;
    #X obj 134 17 inlet;
    #X obj 163 109 unpack f f f f;
    #X obj 132 41 t b a;
    #X obj 98 65 delay 20;
70 #X obj 140 161 *~ 0;
    #X obj 168 182 *~ 0;
    #X obj 195 205 *~ 0;
    #X obj 222 226 *~ 0;
    #X obj 141 127 osc~ 100;
75 #X obj 177 156 osc~ 200;
    #X obj 198 183 osc~ 300;
    #X obj 227 204 osc~ 400;
    #X obj 140 262 outlet~;
    #X msg 99 86 0 0 0 0;
80 #X connect 0 0 2 0;
    #X connect 1 0 4 1;
    #X connect 1 1 5 1;
    #X connect 1 2 6 1;
    #X connect 1 3 7 1;
85 #X connect 2 0 3 0;
    #X connect 2 1 1 0;
    #X connect 3 0 13 0;
    #X connect 4 0 12 0;
    #X connect 5 0 12 0;
90 #X connect 6 0 12 0;
    #X connect 7 0 12 0;
    #X connect 8 0 4 0;
    #X connect 9 0 5 0;
    #X connect 10 0 6 0;
95 #X connect 11 0 7 0;
    #X connect 13 0 1 0;
    #X restore 270 121 pd \$0-1111;
    #N canvas 430 0 450 531 \$0-1000 0;
    #X obj 70 14 inlet;
100 #X obj 143 187 unpack f f f f;
    #X msg 160 115 0;
    #X obj 114 63 delay 20;
    #X obj 70 40 t a b;
    #X obj 69 114 spigot 1;
105 #X msg 114 87 1;
    #X obj 69 134 t b a b;
    #X obj 71 515 outlet~;
    #X obj 69 254 *~;
    #X obj 69 274 *~;
110 #X obj 69 317 -~ 0.25;
    #X obj 70 340 cos~;
    #X obj 72 369 *~;
    #X obj 124 334 +~ 1;
    #X obj 75 390 expr~ tanh($v1);
115 #X obj 73 488 *~;
    #X obj 224 281 * 4;
```

```

#X obj 73 436 expr~ sin($v1);
#X obj 197 356 * 3;
#X obj 196 377 + 1.5;
120 #X obj 69 222 vline~;
#X obj 126 312 *~;
#X obj 153 313 vline~;
#X obj 73 411 *~;
#X obj 103 411 vline~;
125 #X obj 72 465 samphold~;
#X obj 254 359 swap 1;
#X obj 254 389 -;
#X obj 257 448 mtof;
#X obj 255 408 * 60;
130 #X obj 256 505 phasor~;
#X obj 257 467 vline~;
#X obj 259 488 lop~ 25;
#X msg 69 185 1 \, 0 500;
#X obj 68 296 *~ 25;
135 #X obj 256 429 + 80;
#X connect 0 0 4 0;
#X connect 1 1 27 0;
#X connect 1 2 19 0;
#X connect 1 3 17 0;
140 #X connect 2 0 5 1;
#X connect 3 0 6 0;
#X connect 4 0 5 0;
#X connect 4 1 3 0;
#X connect 5 0 7 0;
145 #X connect 6 0 5 1;
#X connect 7 0 34 0;
#X connect 7 1 1 0;
#X connect 7 2 2 0;
#X connect 9 0 10 0;
150 #X connect 9 0 10 1;
#X connect 9 0 22 0;
#X connect 10 0 35 0;
#X connect 11 0 12 0;
#X connect 12 0 13 0;
155 #X connect 13 0 15 0;
#X connect 14 0 13 1;
#X connect 15 0 24 0;
#X connect 16 0 8 0;
#X connect 17 0 23 0;
160 #X connect 18 0 26 0;
#X connect 19 0 20 0;
#X connect 20 0 25 0;
#X connect 21 0 9 0;
#X connect 21 0 9 1;
165 #X connect 21 0 16 1;
#X connect 22 0 14 0;
#X connect 23 0 22 1;
#X connect 24 0 18 0;
#X connect 25 0 24 1;
170 #X connect 26 0 16 0;
#X connect 27 0 28 0;
#X connect 27 1 28 1;
#X connect 28 0 30 0;

```

```
#X connect 29 0 32 0;
175 #X connect 30 0 36 0;
#X connect 31 0 26 1;
#X connect 32 0 33 0;
#X connect 33 0 31 0;
#X connect 34 0 21 0;
180 #X connect 35 0 11 0;
#X connect 36 0 29 0;
#X restore 13 134 pd \ $0-1000;
#N canvas 467 225 450 300 \ $0-0001 0;
#X obj 67 9 inlet;
185 #X obj 77 511 outlet~;
#X obj 143 187 unpack f f f f;
#X msg 160 115 0;
#X obj 114 63 delay 20;
#X obj 70 40 t a b;
190 #X obj 69 114 spigot 1;
#X msg 114 87 1;
#X obj 69 134 t b a b;
#X obj 69 254 *~;
#X obj 69 274 *~;
195 #X obj 69 317 -~ 0.25;
#X obj 70 340 cos~;
#X obj 72 369 *~;
#X obj 124 334 +~ 1;
#X obj 75 390 expr~ tanh($v1);
200 #X obj 73 488 *~;
#X obj 152 275 * 4;
#X obj 73 436 expr~ sin($v1);
#X obj 197 356 * 3;
#X obj 196 377 + 1.5;
205 #X obj 69 222 vline~;
#X obj 126 312 *~;
#X obj 153 313 vline~;
#X obj 73 411 *~;
#X obj 103 411 vline~;
210 #X obj 72 465 samphold~;
#X obj 254 359 swap 1;
#X obj 254 389 -;
#X obj 257 448 mtof;
#X obj 255 408 * 60;
215 #X obj 256 505 phasor~;
#X obj 257 467 vline~;
#X obj 259 488 lop~ 25;
#X msg 69 185 1 \, 0 500;
#X obj 68 296 *~ 25;
220 #X obj 256 429 + 80;
#X connect 0 0 5 0;
#X connect 2 0 17 0;
#X connect 2 1 19 0;
#X connect 2 2 27 0;
225 #X connect 3 0 6 1;
#X connect 4 0 7 0;
#X connect 5 0 6 0;
#X connect 5 1 4 0;
#X connect 6 0 8 0;
230 #X connect 7 0 6 1;
```

```
#X connect 8 0 34 0;
#X connect 8 1 2 0;
#X connect 8 2 3 0;
#X connect 9 0 10 0;
235 #X connect 9 0 10 1;
#X connect 9 0 22 0;
#X connect 10 0 35 0;
#X connect 11 0 12 0;
#X connect 12 0 13 0;
240 #X connect 13 0 15 0;
#X connect 14 0 13 1;
#X connect 15 0 24 0;
#X connect 16 0 1 0;
#X connect 17 0 23 0;
245 #X connect 18 0 26 0;
#X connect 19 0 20 0;
#X connect 20 0 25 0;
#X connect 21 0 9 0;
#X connect 21 0 9 1;
250 #X connect 21 0 16 1;
#X connect 22 0 14 0;
#X connect 23 0 22 1;
#X connect 24 0 18 0;
#X connect 25 0 24 1;
255 #X connect 26 0 16 0;
#X connect 27 0 28 0;
#X connect 27 1 28 1;
#X connect 28 0 30 0;
#X connect 29 0 32 0;
260 #X connect 30 0 36 0;
#X connect 31 0 26 1;
#X connect 32 0 33 0;
#X connect 33 0 31 0;
#X connect 34 0 21 0;
265 #X connect 35 0 11 0;
#X connect 36 0 29 0;
#X restore 175 125 pd \${0}-0001;
#N canvas 191 155 450 428 \${0}-0100 1;
#X obj 103 17 inlet;
270 #X obj 150 340 outlet~;
#X obj 202 339 outlet~;
#X obj 47 227 borgan~;
#X obj 47 128 loadbang;
#X obj 163 109 unpack f f f f;
275 #X obj 98 44 t b a;
#X msg 100 86 0 0 0 0;
#X obj 50 176 unpack f f f;
#X obj 97 227 borgan~;
#X obj 147 227 borgan~;
280 #X obj 197 227 borgan~;
#X obj 247 227 borgan~;
#X obj 297 227 borgan~;
#X obj 48 276 *~;
#X obj 162 153 vline~;
285 #X obj 210 155 vline~;
#X obj 262 155 vline~;
#X obj 160 175 lop~ 25;
```

```
#X obj 210 175 lop~ 25;
#X obj 265 178 lop~ 25;
290 #X obj 98 276 *~;
#X obj 148 276 *~;
#X obj 198 276 *~;
#X obj 248 276 *~;
#X obj 296 276 *~;
295 #X msg 47 153 36 43 24;
#X obj 98 65 delay 456;
#X connect 0 0 6 0;
#X connect 3 0 14 0;
#X connect 4 0 26 0;
300 #X connect 5 0 15 0;
#X connect 5 2 16 0;
#X connect 5 3 17 0;
#X connect 6 0 27 0;
#X connect 6 1 5 0;
305 #X connect 7 0 5 0;
#X connect 8 0 3 0;
#X connect 8 0 9 0;
#X connect 8 1 10 0;
#X connect 8 1 11 0;
310 #X connect 8 2 12 0;
#X connect 8 2 13 0;
#X connect 9 0 21 0;
#X connect 10 0 22 0;
#X connect 11 0 23 0;
315 #X connect 12 0 24 0;
#X connect 13 0 25 0;
#X connect 14 0 1 0;
#X connect 15 0 18 0;
#X connect 16 0 19 0;
320 #X connect 17 0 20 0;
#X connect 18 0 14 1;
#X connect 18 0 21 1;
#X connect 19 0 22 1;
#X connect 19 0 23 1;
325 #X connect 20 0 25 1;
#X connect 20 0 24 1;
#X connect 21 0 2 0;
#X connect 22 0 1 0;
#X connect 23 0 2 0;
330 #X connect 24 0 1 0;
#X connect 25 0 2 0;
#X connect 26 0 8 0;
#X connect 27 0 7 0;
#X restore 82 126 pd \${0-0100};
335 #N canvas 191 155 450 428 \${0-0010 1;
#X obj 103 17 inlet;
#X obj 150 340 outlet~;
#X obj 202 339 outlet~;
#X obj 47 227 borgan~;
340 #X obj 47 128 loadbang;
#X obj 163 109 unpack f f f f;
#X obj 98 44 t b a;
#X msg 100 86 0 0 0 0;
#X obj 50 176 unpack f f f;
```

```

345  #X obj 97 227 borgan~;
    #X obj 147 227 borgan~;
    #X obj 197 227 borgan~;
    #X obj 247 227 borgan~;
    #X obj 297 227 borgan~;
350  #X obj 48 276 *~;
    #X obj 162 153 vline~;
    #X obj 210 155 vline~;
    #X obj 262 155 vline~;
    #X obj 160 175 lop~ 25;
355  #X obj 210 175 lop~ 25;
    #X obj 265 178 lop~ 25;
    #X obj 98 276 *~;
    #X obj 148 276 *~;
    #X obj 198 276 *~;
360  #X obj 248 276 *~;
    #X obj 296 276 *~;
    #X msg 47 153 24 41 36;
    #X obj 98 65 delay 229;
    #X connect 0 0 6 0;
365  #X connect 3 0 14 0;
    #X connect 4 0 26 0;
    #X connect 5 0 15 0;
    #X connect 5 1 16 0;
    #X connect 5 3 17 0;
370  #X connect 6 0 27 0;
    #X connect 6 1 5 0;
    #X connect 7 0 5 0;
    #X connect 8 0 3 0;
    #X connect 8 0 9 0;
375  #X connect 8 1 10 0;
    #X connect 8 1 11 0;
    #X connect 8 2 12 0;
    #X connect 8 2 13 0;
    #X connect 9 0 21 0;
380  #X connect 10 0 22 0;
    #X connect 11 0 23 0;
    #X connect 12 0 24 0;
    #X connect 13 0 25 0;
    #X connect 14 0 1 0;
385  #X connect 15 0 18 0;
    #X connect 16 0 19 0;
    #X connect 17 0 20 0;
    #X connect 18 0 14 1;
    #X connect 18 0 21 1;
390  #X connect 19 0 22 1;
    #X connect 19 0 23 1;
    #X connect 20 0 25 1;
    #X connect 20 0 24 1;
    #X connect 21 0 2 0;
395  #X connect 22 0 1 0;
    #X connect 23 0 2 0;
    #X connect 24 0 1 0;
    #X connect 25 0 2 0;
    #X connect 26 0 8 0;
400  #X connect 27 0 7 0;
    #X restore 102 106 pd \ $0-0010;

```

```
#X connect 0 0 3 0;
#X connect 3 1 5 0;
#X connect 3 2 7 0;
405 #X connect 3 4 8 0;
#X connect 3 8 6 0;
#X connect 3 15 4 0;
#X connect 4 0 2 0;
#X connect 4 0 1 0;
410 #X connect 5 0 1 0;
#X connect 5 0 2 0;
#X connect 6 0 1 0;
#X connect 6 0 2 0;
#X connect 7 0 1 0;
415 #X connect 7 1 2 0;
#X connect 8 0 1 0;
#X connect 8 1 2 0;
#X restore 364 686 pd \$0-audio;
#X obj 427 710 *~ 0.1;
420 #X obj 337 710 *~ 0.1;
#X floatatom 505 662 5 0 0 0 - - -;
#X obj 279 242 f;
#X obj 279 221 sel 1;
#X obj 279 198 change;
425 #X obj 101 561 list prepend set;
#X obj 102 584 list trim;
#X msg 106 618 6 21 38 118 27 0 0 0 1 1 1;
#X msg 153 432 0 0 0 1 1 1;
#X msg 171 409 0 1 1 0 0 1;
430 #X msg 186 387 1 0 1 0 1 0;
#X msg 201 368 1 1 0 1 0 0;
#X obj 374 166 t f f f;
#X msg 99 374 1 1 1 10;
#X msg 79 415 1 10 1 1;
435 #X msg 89 394 1 1 10 1;
#X obj 96 484 list append;
#X obj 97 510 list prepend 6;
#X obj 69 347 sel 0 1 3 2;
#X obj 155 348 sel 0 1 3 2;
440 #X msg 69 435 1 1 1 1;
#X obj 418 147 mod 96;
#X obj 344 104 metro 19;
#X connect 0 0 1 0;
#X connect 0 1 1 1;
445 #X connect 0 1 4 0;
#X connect 0 2 1 2;
#X connect 1 0 2 0;
#X connect 2 0 3 0;
#X connect 3 0 5 0;
450 #X connect 5 0 34 1;
#X connect 5 1 34 2;
#X connect 5 2 34 3;
#X connect 5 3 34 4;
#X connect 5 4 34 5;
455 #X connect 5 5 34 6;
#X connect 5 6 34 7;
#X connect 5 7 34 8;
#X connect 5 8 32 0;
```

```
#X connect 7 0 8 0;
460 #X connect 7 0 71 0;
#X connect 8 0 80 0;
#X connect 11 0 38 0;
#X connect 12 0 28 7;
#X connect 13 0 27 0;
465 #X connect 13 1 14 0;
#X connect 13 2 15 0;
#X connect 13 3 16 0;
#X connect 13 4 17 0;
#X connect 13 5 18 0;
470 #X connect 13 6 19 0;
#X connect 13 7 20 0;
#X connect 14 0 26 0;
#X connect 15 0 25 0;
#X connect 16 0 24 0;
475 #X connect 17 0 23 0;
#X connect 18 0 22 0;
#X connect 19 0 21 0;
#X connect 20 0 12 0;
#X connect 21 0 28 6;
480 #X connect 22 0 28 5;
#X connect 23 0 28 4;
#X connect 24 0 28 3;
#X connect 25 0 28 2;
#X connect 26 0 28 1;
485 #X connect 27 0 29 0;
#X connect 28 0 30 0;
#X connect 29 0 28 0;
#X connect 29 1 31 0;
#X connect 30 0 39 0;
490 #X connect 31 0 30 1;
#X connect 32 0 33 0;
#X connect 33 0 63 0;
#X connect 34 0 35 0;
#X connect 36 0 10 0;
495 #X connect 37 0 81 0;
#X connect 38 0 13 0;
#X connect 38 1 61 1;
#X connect 39 0 41 0;
#X connect 39 1 42 0;
500 #X connect 39 2 43 0;
#X connect 39 3 44 0;
#X connect 39 4 47 0;
#X connect 39 5 48 0;
#X connect 39 6 49 0;
505 #X connect 39 7 50 0;
#X connect 41 0 45 0;
#X connect 42 0 45 1;
#X connect 43 0 45 2;
#X connect 44 0 45 3;
510 #X connect 45 0 56 0;
#X connect 46 0 54 0;
#X connect 47 0 55 0;
#X connect 48 0 46 0;
#X connect 49 0 51 0;
515 #X connect 50 0 52 0;
```

```

#X connect 51 0 53 0;
#X connect 52 0 53 1;
#X connect 53 0 54 1;
#X connect 54 0 55 1;
520 #X connect 55 0 56 1;
#X connect 55 0 60 0;
#X connect 56 0 57 0;
#X connect 57 0 59 0;
#X connect 57 1 58 0;
525 #X connect 58 0 40 1;
#X connect 59 0 40 0;
#X connect 61 0 34 0;
#X connect 62 0 61 0;
#X connect 63 0 62 0;
530 #X connect 64 0 65 0;
#X connect 65 0 66 0;
#X connect 67 0 75 1;
#X connect 68 0 75 1;
#X connect 69 0 75 1;
535 #X connect 70 0 75 1;
#X connect 71 0 11 0;
#X connect 72 0 75 0;
#X connect 73 0 75 0;
#X connect 74 0 75 0;
540 #X connect 75 0 76 0;
#X connect 76 0 6 0;
#X connect 77 0 79 0;
#X connect 77 1 73 0;
#X connect 77 2 74 0;
545 #X connect 77 3 72 0;
#X connect 78 0 67 0;
#X connect 78 1 68 0;
#X connect 78 2 69 0;
#X connect 78 3 70 0;
550 #X connect 79 0 75 0;
#X connect 80 0 7 1;
#X connect 81 0 7 0;

```

## 26 -/reflex-onescreen.sh

```

#!/bin/bash
pd -stderr -nrt -r 48000 -oss -channels 2 -open reflex.pd 2>&1 |
./reflex
#grep "^reflex: " |
5 #sed "s/^reflex: //" |
# |
#ppmtoy4m -S 444 -F 25:1 |
#y4mscaler -O preset=dvd -O yscale=1/1 |
#mpeg2enc -f 8 -q 2 -b 8000 -o reflex.m2v

```

## 27 -/reflex.pd

```

#N canvas 0 0 563 482 10;
#X obj 35 206 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 1
1;
#X obj 296 35 dac~;

```

```

5  #X msg 56 193 \; pd dsp 1;
   #X obj 446 199 + 0.01;
   #X obj 353 199 + 0.01;
   #X obj 398 198 + 0.01;
   #X obj 160 149 pack f f;
10  #X obj 239 254 route 2 3 4 5 6 8 9 12 13;
   #X obj 160 125 swap;
   #X obj 160 92 ctlin;
   #X obj 244 282 vsl 15 128 0 127 0 0 empty empty empty 0 -9 0 10 -262144
   -1 -1 0 1;
15  #X obj 264 282 vsl 15 128 0 127 0 0 empty empty empty 0 -9 0 10 -262144
   -1 -1 0 1;
   #X obj 283 282 vsl 15 128 0 127 0 0 empty empty empty 0 -9 0 10 -262144
   -1 -1 0 1;
   #X obj 303 282 vsl 15 128 0 127 0 0 empty empty empty 0 -9 0 10 -262144
   -1 -1 0 1;
20  #X obj 323 282 vsl 15 128 0 127 0 0 empty empty empty 0 -9 0 10 -262144
   -1 -1 0 1;
   #X obj 343 282 vsl 15 128 0 127 0 0 empty empty empty 0 -9 0 10 -262144
   -1 -1 0 1;
25  #X obj 363 282 vsl 15 128 0 127 0 0 empty empty empty 0 -9 0 10 -262144
   -1 -1 0 1;
   #X obj 383 282 vsl 15 128 0 127 0 0 empty empty empty 0 -9 0 10 -262144
   -1 -1 0 1;
   #X obj 403 282 vsl 15 128 0 127 0 0 empty empty empty 0 -9 0 10 -262144
   -1 -1 0 1;
30  #X obj 124 123 + 1;
   #X obj 122 188 pack f f;
   #X obj 122 216 route 2 3 4 5 6 8 9 12 13;
   #X obj 122 164 swap;
35  #X obj 124 143 / 128;
   #X obj 31 268 pack f f f f f;
   #X msg 32 291 5 \$2 \$3 \$4 \$5;
   #X obj 33 227 metro 40;
   #X obj 27 322 print reflex;
40  #N canvas 738 261 450 300 \$0-fb-stuff 0;
   #X obj 106 209 expr~ tanh($v1);
   #X obj 230 210 expr~ tanh($v1);
   #X obj 104 269 outlet~;
   #X obj 224 269 outlet~;
45  #X obj 98 50 delread~ \$0-fb-l 1000;
   #X obj 238 50 delread~ \$0-fb-r 1125;
   #X obj 121 241 delwrite~ \$0-fb-l 2000;
   #X obj 271 240 delwrite~ \$0-fb-r 2000;
   #X obj 104 182 vcf~ 5;
50  #X obj 104 132 vcf~ 5;
   #X obj 154 111 sig~;
   #X obj 154 88 mtof;
   #X obj 156 134 lop~ 25;
   #X obj 153 20 inlet;
55  #X obj 108 91 noise~;
   #X obj 155 68 * 128;
   #X obj 29 156 *~ 0.8;
   #X obj 104 158 expr~ tanh($v1*2)*2;
   #X obj 234 182 vcf~ 5;
60  #X obj 234 132 vcf~ 5;
   #X obj 234 158 expr~ tanh($v1*2)*2;

```

```
#X obj 248 91 noise~;
#X obj 389 156 *~ 0.8;
#X obj 162 269 outlet~;
65 #X obj 165 179 *~ 0.2;
#X connect 0 0 2 0;
#X connect 0 0 6 0;
#X connect 0 0 23 0;
#X connect 1 0 3 0;
70 #X connect 1 0 7 0;
#X connect 1 0 2 0;
#X connect 4 0 9 0;
#X connect 4 0 16 0;
#X connect 5 0 19 0;
75 #X connect 5 0 22 0;
#X connect 8 0 0 0;
#X connect 9 0 17 0;
#X connect 10 0 12 0;
#X connect 11 0 10 0;
80 #X connect 12 0 9 1;
#X connect 12 0 8 1;
#X connect 12 0 19 1;
#X connect 12 0 18 1;
#X connect 13 0 15 0;
85 #X connect 14 0 9 0;
#X connect 15 0 11 0;
#X connect 16 0 0 0;
#X connect 17 0 8 0;
#X connect 17 0 24 0;
90 #X connect 18 0 1 0;
#X connect 19 0 20 0;
#X connect 20 0 18 0;
#X connect 20 0 24 0;
#X connect 21 0 19 0;
95 #X connect 22 0 1 0;
#X connect 24 0 8 0;
#X connect 24 0 18 0;
#X restore 250 11 pd \ $0-fb-stuff;
#X obj 263 138 dbtorms;
100 #X obj 334 138 dbtorms;
#X obj 404 138 dbtorms;
#X obj 474 138 dbtorms;
#X obj 336 77 bp~ 300 10;
#X obj 265 77 bp~ 100 10;
105 #X obj 406 77 bp~ 1000 10;
#X obj 476 77 bp~ 3000 10;
#X obj 264 161 expr pow($f1 \, 2);
#X obj 343 181 expr pow($f1 \, 2);
#X obj 373 161 expr pow($f1 \, 2);
110 #X obj 450 180 expr pow($f1 \, 2);
#X obj 265 112 env~ 4096;
#X obj 336 112 env~ 4096;
#X obj 406 112 env~ 4096;
#X obj 476 112 env~ 4096;
115 #X floatatom 198 12 5 0 0 0 - - -;
#X obj 288 195 + 0.01;
#X connect 0 0 26 0;
#X connect 3 0 24 4;
```

```
120 #X connect 4 0 24 2;
    #X connect 5 0 24 3;
    #X connect 6 0 7 0;
    #X connect 7 0 10 0;
    #X connect 7 1 11 0;
    #X connect 7 2 12 0;
125 #X connect 7 3 13 0;
    #X connect 7 4 14 0;
    #X connect 7 5 15 0;
    #X connect 7 6 16 0;
    #X connect 7 7 17 0;
130 #X connect 7 8 18 0;
    #X connect 8 0 6 0;
    #X connect 8 1 6 1;
    #X connect 9 0 8 0;
    #X connect 9 0 19 0;
135 #X connect 9 1 8 1;
    #X connect 9 1 22 1;
    #X connect 19 0 23 0;
    #X connect 20 0 21 0;
    #X connect 21 7 28 0;
140 #X connect 22 0 20 0;
    #X connect 22 1 20 1;
    #X connect 23 0 22 0;
    #X connect 24 0 25 0;
    #X connect 25 0 27 0;
145 #X connect 26 0 24 0;
    #X connect 28 0 33 0;
    #X connect 28 0 34 0;
    #X connect 28 0 35 0;
    #X connect 28 0 36 0;
150 #X connect 28 1 1 0;
    #X connect 28 2 1 1;
    #X connect 29 0 37 0;
    #X connect 30 0 38 0;
    #X connect 31 0 39 0;
155 #X connect 32 0 40 0;
    #X connect 33 0 42 0;
    #X connect 34 0 41 0;
    #X connect 35 0 43 0;
    #X connect 36 0 44 0;
160 #X connect 37 0 46 0;
    #X connect 38 0 4 0;
    #X connect 39 0 5 0;
    #X connect 40 0 3 0;
    #X connect 41 0 29 0;
165 #X connect 42 0 30 0;
    #X connect 43 0 31 0;
    #X connect 44 0 32 0;
    #X connect 45 0 28 0;
    #X connect 46 0 24 1;
```

## 28 reflex.pd

```
#N canvas 0 0 1099 692 10;
#X obj 28 28 netreceive 6666;
#X obj 24 642 netsend;
```

```

#X msg 24 593 connect localhost 5555;
5 #X obj 101 517 list prepend send;
#X obj 101 539 list trim;
#X obj 24 571 spigot 1;
#X obj 107 615 == 0;
#X obj 42 527 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
10 1;
#X obj 24 506 loadbang;
#X obj 24 547 metro 1000;
#X obj 101 565 spigot 0;
#X obj 207 553 loadbang;
15 #X obj 206 580 delay 1000;
#X msg 207 607 \; pd dsp 1;
#X obj 256 36 ctlin;
#X obj 241 73 pack f f f;
#X msg 301 104 set \ $3 \ $2 \ $1;
20 #X msg 301 131 15 7 36;
#X obj 239 130 route 15;
#X obj 239 158 route 66 67 68 69 70 71 72 73;
#X obj 186 356 sel 0 1;
#X obj 182 213 >= 64;
25 #X msg 186 381 XY Off;
#X msg 205 405 XY On;
#X obj 186 439 t a;
#X msg 239 106 \ $3 \ $2 \ $1;
#X obj 246 356 sel 0 1;
30 #X obj 250 180 >= 64;
#X obj 246 439 t a;
#X obj 306 356 sel 0 1;
#X obj 290 183 >= 64;
#X obj 306 439 t a;
35 #X obj 366 356 sel 0 1;
#X obj 362 183 >= 64;
#X obj 366 439 t a;
#X obj 426 356 sel 0 1;
#X obj 381 203 >= 64;
40 #X obj 426 439 t a;
#X obj 486 356 sel 0 1;
#X obj 477 260 >= 64;
#X obj 486 439 t a;
#X msg 246 381 XZ Off;
45 #X msg 265 405 XZ On;
#X msg 306 381 XW Off;
#X msg 325 405 XW On;
#X msg 366 381 YZ Off;
#X msg 385 405 YZ On;
50 #X msg 426 381 YW Off;
#X msg 445 405 YW On;
#X msg 486 381 ZW Off;
#X msg 505 405 ZW On;
#X obj 184 482 t a;
55 #X obj 6 468 dac~;
#X obj 558 350 route 16 17 18 19;
#X obj 577 368 t b f;
#X obj 560 391 pack 1 1 1 1;
#X obj 618 366 t b f;
60 #X obj 659 369 t b f;

```

```

#X msg 563 421 Shape \$1 \$2 \$3 \$4;
#X obj 710 203 t b f;
#X obj 696 235 pack 1 1 1 1;
#X obj 750 203 t b f;
65 #X obj 790 203 t b f;
#X obj 692 146 route 20 21 22 23;
#X obj 697 265 unpack f f f f;
#X obj 696 293 *;
#X obj 726 293 *;
70 #X obj 756 293 *;
#X obj 701 346 pack f f f;
#X msg 698 400 Move \$1 \$2 \$3;
#X obj 695 175 - 64;
#X obj 727 176 - 64;
75 #X obj 765 175 - 64;
#X obj 698 316 / 1000;
#X obj 748 316 / 1000;
#X obj 798 316 / 1000;
#X obj 25 442 delwrite~ \$0-l 10000;
80 #X obj 58 467 delwrite~ \$0-r 10000;
#X obj 38 345 limiter~ 50 1;
#X obj 73 369 z~ 64;
#X obj 25 367 z~ 64;
#X obj 25 401 *~;
85 #X obj 58 400 *~;
#X obj 25 319 hip~ 1;
#X obj 71 320 hip~ 1;
#X obj 26 125 vd~ \$0-l;
#X obj 71 147 vd~ \$0-r;
90 #X obj 28 54 sig~ 444;
#N canvas 0 170 1072 376 \$0-filterbank 0;
#X obj 415 50 inlet~;
#X obj 489 51 inlet~;
#X obj 298 340 outlet~;
95 #X obj 434 339 outlet~;
#X obj 503 339 outlet;
#X obj 619 17 inlet;
#X obj 132 162 vcf~ 10;
#X obj 133 135 *~ 1;
100 #X obj 133 224 -~ 0.25;
#X obj 133 246 cos~;
#X obj 21 134 *~ 1;
#X obj 33 159 vcf~ 10;
#X obj 77 129 sig~ 50;
105 #X obj 34 221 -~ 0.25;
#X obj 34 243 cos~;
#X obj 302 162 vcf~ 10;
#X obj 303 135 *~ 1;
#X obj 303 224 -~ 0.25;
110 #X obj 303 246 cos~;
#X obj 191 134 *~ 1;
#X obj 203 159 vcf~ 10;
#X obj 204 221 -~ 0.25;
#X obj 204 243 cos~;
115 #X obj 472 162 vcf~ 10;
#X obj 473 135 *~ 1;
#X obj 473 224 -~ 0.25;

```

```

#X obj 473 246 cos ~;
#X obj 361 134 *~ 1;
120 #X obj 373 159 vcf~ 10;
#X obj 374 221 -~ 0.25;
#X obj 374 243 cos ~;
#X obj 642 162 vcf~ 10;
#X obj 643 135 *~ 1;
125 #X obj 643 224 -~ 0.25;
#X obj 643 246 cos ~;
#X obj 531 134 *~ 1;
#X obj 543 159 vcf~ 10;
#X obj 544 221 -~ 0.25;
130 #X obj 544 243 cos ~;
#X obj 247 129 sig~ 100;
#X obj 402 132 sig~ 200;
#X obj 587 129 sig~ 400;
#X obj 812 162 vcf~ 10;
135 #X obj 813 135 *~ 1;
#X obj 813 224 -~ 0.25;
#X obj 813 246 cos ~;
#X obj 701 134 *~ 1;
#X obj 713 159 vcf~ 10;
140 #X obj 714 221 -~ 0.25;
#X obj 714 243 cos ~;
#X obj 992 162 vcf~ 10;
#X obj 993 135 *~ 1;
#X obj 993 224 -~ 0.25;
145 #X obj 993 246 cos ~;
#X obj 881 134 *~ 1;
#X obj 893 159 vcf~ 10;
#X obj 894 221 -~ 0.25;
#X obj 894 243 cos ~;
150 #X obj 757 129 sig~ 800;
#X obj 924 130 sig~ 1600;
#X obj 621 36 route 0 1 2 3 4 5;
#X obj 599 68 f;
#X obj 629 68 f;
155 #X obj 659 68 f;
#X obj 689 68 f;
#X obj 719 65 f;
#X obj 750 65 f;
#X obj 301 314 *~ 0.5;
160 #X obj 434 314 *~ 0.5;
#X connect 0 0 10 0;
#X connect 0 0 19 0;
#X connect 0 0 27 0;
#X connect 0 0 35 0;
165 #X connect 0 0 46 0;
#X connect 0 0 54 0;
#X connect 1 0 7 0;
#X connect 1 0 16 0;
#X connect 1 0 24 0;
170 #X connect 1 0 32 0;
#X connect 1 0 43 0;
#X connect 1 0 51 0;
#X connect 5 0 60 0;
#X connect 6 0 8 0;

```

```
175  #X connect 7 0 6 0;
      #X connect 8 0 9 0;
      #X connect 9 0 68 0;
      #X connect 10 0 11 0;
      #X connect 11 0 13 0;
180  #X connect 12 0 11 1;
      #X connect 12 0 6 1;
      #X connect 13 0 14 0;
      #X connect 14 0 67 0;
      #X connect 15 0 17 0;
185  #X connect 16 0 15 0;
      #X connect 17 0 18 0;
      #X connect 18 0 68 0;
      #X connect 19 0 20 0;
      #X connect 20 0 21 0;
190  #X connect 21 0 22 0;
      #X connect 22 0 67 0;
      #X connect 23 0 25 0;
      #X connect 24 0 23 0;
      #X connect 25 0 26 0;
195  #X connect 26 0 68 0;
      #X connect 27 0 28 0;
      #X connect 28 0 29 0;
      #X connect 29 0 30 0;
      #X connect 30 0 67 0;
200  #X connect 31 0 33 0;
      #X connect 32 0 31 0;
      #X connect 33 0 34 0;
      #X connect 34 0 68 0;
      #X connect 35 0 36 0;
205  #X connect 36 0 37 0;
      #X connect 37 0 38 0;
      #X connect 38 0 67 0;
      #X connect 39 0 20 1;
      #X connect 39 0 15 1;
210  #X connect 40 0 28 1;
      #X connect 40 0 23 1;
      #X connect 41 0 36 1;
      #X connect 41 0 31 1;
      #X connect 42 0 44 0;
215  #X connect 43 0 42 0;
      #X connect 44 0 45 0;
      #X connect 45 0 68 0;
      #X connect 46 0 47 0;
      #X connect 47 0 48 0;
220  #X connect 48 0 49 0;
      #X connect 49 0 67 0;
      #X connect 50 0 52 0;
      #X connect 51 0 50 0;
      #X connect 52 0 53 0;
225  #X connect 53 0 68 0;
      #X connect 54 0 55 0;
      #X connect 55 0 56 0;
      #X connect 56 0 57 0;
      #X connect 57 0 67 0;
230  #X connect 58 0 47 1;
      #X connect 58 0 42 1;
```

```

#X connect 59 0 55 1;
#X connect 59 0 50 1;
#X connect 60 0 61 0;
235 #X connect 60 1 62 0;
#X connect 60 2 63 0;
#X connect 60 3 64 0;
#X connect 60 4 65 0;
#X connect 60 5 66 0;
240 #X connect 61 0 10 1;
#X connect 61 0 7 1;
#X connect 62 0 19 1;
#X connect 62 0 16 1;
#X connect 63 0 27 1;
245 #X connect 63 0 24 1;
#X connect 64 0 35 1;
#X connect 64 0 32 1;
#X connect 65 0 46 1;
#X connect 65 0 43 1;
250 #X connect 66 0 54 1;
#X connect 66 0 51 1;
#X connect 67 0 2 0;
#X connect 68 0 3 0;
#X restore 25 273 pd \ $0-filterbank ;
255 #X msg 302 247 2 \ $1;
#X msg 254 316 1 \ $1;
#X msg 333 322 3 \ $1;
#X msg 393 271 4 \ $1;
#X msg 429 330 5 \ $1;
260 #X msg 218 253 0 \ $1;
#X obj 218 206 moses 64;
#X msg 217 226 0;
#X obj 263 227 - 64;
#X obj 35 220 noise ~;
265 #X obj 35 245 *~ 1e-06;
#X obj 91 221 noise ~;
#X obj 91 246 *~ 1e-06;
#X obj 254 274 moses 64;
#X msg 253 294 0;
270 #X obj 299 295 - 64;
#X obj 302 202 moses 64;
#X msg 301 222 0;
#X obj 347 223 - 64;
#X obj 334 274 moses 64;
275 #X msg 333 294 0;
#X obj 379 295 - 64;
#X obj 392 225 moses 64;
#X msg 391 245 0;
#X obj 437 246 - 64;
280 #X obj 432 285 moses 64;
#X msg 431 305 0;
#X obj 477 306 - 64;
#X obj 262 248 / 32;
#X obj 298 316 / 32;
285 #X obj 346 244 / 32;
#X obj 378 316 / 32;
#X obj 436 267 / 32;
#X obj 476 327 / 32;

```

```

#X obj 602 47 f 0;
290 #X obj 675 38 + 1;
#X obj 602 2 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0 1
;
#X obj 602 23 metro 222;
#X obj 675 59 mod 8;
295 #X obj 297 595 s \ $0-controllermap;
#X obj 292 530 loadbang;
#X obj 331 59 table \ $0-controllernow 128;
#X obj 333 36 s \ $0-controllernow;
#X obj 581 488 until;
300 #X msg 581 467 128;
#X obj 581 513 f 0;
#X obj 638 504 + 1;
#X obj 638 528 mod 128;
#X obj 581 566 tabread \ $0-controllernow;
305 #X obj 581 539 t f f;
#X obj 716 597 +;
#X obj 731 566 * 128;
#X obj 284 674 table \ $0-controllers 1024;
#X obj 581 619 tabwrite \ $0-controllers;
310 #X msg 286 640 const -1;
#X obj 286 658 s \ $0-controllers;
#X obj 582 444 t b f;
#X obj 854 440 / 16;
#X obj 855 466 int;
315 #X obj 841 91 hradio 15 1 0 8 empty empty empty 0 -8 0 10 -262144 -1
-1 1;
#X obj 800 508 f;
#X obj 826 383 route 7 48;
#X obj 795 414 >= 64;
320 #X obj 795 434 change;
#X obj 796 457 sel 1;
#X obj 841 63 hradio 15 1 0 8 empty empty empty 0 -8 0 10 -262144 -1
-1 7;
#X obj 533 111 until;
325 #X msg 533 87 128;
#X obj 533 67 t b f;
#X obj 579 82 * 128;
#X obj 533 137 f 0;
#X obj 590 118 + 1;
330 #X obj 590 142 mod 128;
#X obj 533 163 t f f;
#X obj 562 183 tabread \ $0-controllermap;
#X obj 531 207 spigot;
#X obj 532 230 t f f;
335 #X obj 533 252 +;
#X obj 535 271 tabread \ $0-controllers;
#X obj 535 292 moses -0.5;
#X obj 534 313 swap;
#X obj 534 334 pack f f;
340 #X obj 296 615 table \ $0-controllermap 128;
#X obj 29 76 *~ 16;
#X msg 298 555 const 0 \, 16 1 \, 17 1 \, 18 1 \, 19 1 \, 66 1 \, 67
1 \, 68 1 \, 69 1 \, 70 1 \, 71 1;
#X connect 1 0 6 0;
345 #X connect 1 0 10 1;

```

```
#X connect 2 0 1 0;
#X connect 3 0 4 0;
#X connect 4 0 10 0;
#X connect 5 0 2 0;
350 #X connect 6 0 5 1;
#X connect 7 0 9 0;
#X connect 8 0 9 0;
#X connect 9 0 5 0;
#X connect 10 0 1 0;
355 #X connect 11 0 12 0;
#X connect 12 0 13 0;
#X connect 14 0 15 0;
#X connect 14 1 15 1;
#X connect 14 2 15 2;
360 #X connect 15 0 16 0;
#X connect 15 0 25 0;
#X connect 16 0 17 0;
#X connect 18 0 19 0;
#X connect 18 0 131 0;
365 #X connect 19 0 21 0;
#X connect 19 0 95 0;
#X connect 19 1 27 0;
#X connect 19 1 102 0;
#X connect 19 2 30 0;
370 #X connect 19 2 105 0;
#X connect 19 3 33 0;
#X connect 19 3 108 0;
#X connect 19 4 36 0;
#X connect 19 4 111 0;
375 #X connect 19 5 39 0;
#X connect 19 5 114 0;
#X connect 19 8 53 0;
#X connect 20 0 22 0;
#X connect 20 1 23 0;
380 #X connect 21 0 20 0;
#X connect 22 0 24 0;
#X connect 23 0 24 0;
#X connect 24 0 51 0;
#X connect 25 0 18 0;
385 #X connect 26 0 41 0;
#X connect 26 1 42 0;
#X connect 27 0 26 0;
#X connect 28 0 51 0;
#X connect 29 0 43 0;
390 #X connect 29 1 44 0;
#X connect 30 0 29 0;
#X connect 31 0 51 0;
#X connect 32 0 45 0;
#X connect 32 1 46 0;
395 #X connect 33 0 32 0;
#X connect 34 0 51 0;
#X connect 35 0 47 0;
#X connect 35 1 48 0;
#X connect 36 0 35 0;
400 #X connect 37 0 51 0;
#X connect 38 0 49 0;
#X connect 38 1 50 0;
```

```
#X connect 39 0 38 0;
#X connect 40 0 51 0;
405 #X connect 41 0 28 0;
#X connect 42 0 28 0;
#X connect 43 0 31 0;
#X connect 44 0 31 0;
#X connect 45 0 34 0;
410 #X connect 46 0 34 0;
#X connect 47 0 37 0;
#X connect 48 0 37 0;
#X connect 49 0 40 0;
#X connect 50 0 40 0;
415 #X connect 51 0 3 0;
#X connect 53 0 55 0;
#X connect 53 1 54 0;
#X connect 53 2 56 0;
#X connect 53 3 57 0;
420 #X connect 53 4 63 0;
#X connect 54 0 55 0;
#X connect 54 1 55 1;
#X connect 55 0 58 0;
#X connect 56 0 55 0;
425 #X connect 56 1 55 2;
#X connect 57 0 55 0;
#X connect 57 1 55 3;
#X connect 58 0 51 0;
#X connect 59 0 60 0;
430 #X connect 59 1 60 1;
#X connect 60 0 64 0;
#X connect 61 0 60 0;
#X connect 61 1 60 2;
#X connect 62 0 60 0;
435 #X connect 62 1 60 3;
#X connect 63 0 70 0;
#X connect 63 1 71 0;
#X connect 63 2 72 0;
#X connect 63 3 62 0;
440 #X connect 63 4 150 0;
#X connect 64 0 65 0;
#X connect 64 1 66 0;
#X connect 64 2 67 0;
#X connect 64 3 67 1;
445 #X connect 64 3 66 1;
#X connect 64 3 65 1;
#X connect 65 0 73 0;
#X connect 66 0 74 0;
#X connect 67 0 75 0;
450 #X connect 68 0 69 0;
#X connect 69 0 51 0;
#X connect 70 0 60 0;
#X connect 71 0 59 0;
#X connect 72 0 61 0;
455 #X connect 73 0 68 0;
#X connect 74 0 68 1;
#X connect 75 0 68 2;
#X connect 78 0 81 1;
#X connect 78 0 82 0;
```

```
460  #X connect 79 0 82 1;
      #X connect 80 0 81 0;
      #X connect 81 0 76 0;
      #X connect 81 0 52 0;
      #X connect 82 0 77 0;
465  #X connect 82 0 52 1;
      #X connect 83 0 80 0;
      #X connect 83 0 78 0;
      #X connect 84 0 79 0;
      #X connect 84 0 78 1;
470  #X connect 85 0 88 0;
      #X connect 86 0 88 1;
      #X connect 87 0 172 0;
      #X connect 88 0 83 0;
      #X connect 88 1 84 0;
475  #X connect 89 0 88 2;
      #X connect 90 0 88 2;
      #X connect 91 0 88 2;
      #X connect 92 0 88 2;
      #X connect 93 0 88 2;
480  #X connect 94 0 88 2;
      #X connect 95 0 96 0;
      #X connect 95 1 97 0;
      #X connect 96 0 94 0;
      #X connect 97 0 117 0;
485  #X connect 98 0 99 0;
      #X connect 99 0 88 0;
      #X connect 100 0 101 0;
      #X connect 101 0 88 1;
      #X connect 102 0 103 0;
490  #X connect 102 1 104 0;
      #X connect 103 0 90 0;
      #X connect 104 0 118 0;
      #X connect 105 0 106 0;
      #X connect 105 1 107 0;
495  #X connect 106 0 89 0;
      #X connect 107 0 119 0;
      #X connect 108 0 109 0;
      #X connect 108 1 110 0;
      #X connect 109 0 91 0;
500  #X connect 110 0 120 0;
      #X connect 111 0 112 0;
      #X connect 111 1 113 0;
      #X connect 112 0 92 0;
      #X connect 113 0 121 0;
505  #X connect 114 0 115 0;
      #X connect 114 1 116 0;
      #X connect 115 0 93 0;
      #X connect 116 0 122 0;
      #X connect 117 0 94 0;
510  #X connect 118 0 90 0;
      #X connect 119 0 89 0;
      #X connect 120 0 91 0;
      #X connect 121 0 92 0;
      #X connect 122 0 93 0;
515  #X connect 123 0 124 0;
      #X connect 123 0 154 0;
```

```

#X connect 123 0 157 0;
#X connect 124 0 127 0;
#X connect 125 0 126 0;
520 #X connect 126 0 123 0;
#X connect 127 0 123 1;
#X connect 129 0 173 0;
#X connect 129 0 143 0;
#X connect 132 0 134 0;
525 #X connect 133 0 132 0;
#X connect 134 0 135 0;
#X connect 134 0 138 0;
#X connect 135 0 136 0;
#X connect 136 0 134 1;
530 #X connect 137 0 142 0;
#X connect 138 0 137 0;
#X connect 138 1 139 0;
#X connect 139 0 142 1;
#X connect 140 0 139 1;
535 #X connect 143 0 144 0;
#X connect 145 0 133 0;
#X connect 145 1 140 0;
#X connect 146 0 147 0;
#X connect 147 0 148 0;
540 #X connect 147 0 149 1;
#X connect 149 0 145 0;
#X connect 150 0 151 0;
#X connect 150 1 146 0;
#X connect 151 0 152 0;
545 #X connect 152 0 153 0;
#X connect 153 0 149 0;
#X connect 155 0 159 0;
#X connect 156 0 155 0;
#X connect 157 0 156 0;
550 #X connect 157 1 158 0;
#X connect 158 0 166 1;
#X connect 159 0 160 0;
#X connect 159 0 162 0;
#X connect 160 0 161 0;
555 #X connect 161 0 159 1;
#X connect 162 0 164 0;
#X connect 162 1 163 0;
#X connect 163 0 164 1;
#X connect 164 0 165 0;
560 #X connect 165 0 166 0;
#X connect 165 1 169 1;
#X connect 166 0 167 0;
#X connect 167 0 168 0;
#X connect 168 1 169 0;
565 #X connect 169 0 170 0;
#X connect 169 1 170 1;
#X connect 170 0 19 0;
#X connect 172 0 85 0;
#X connect 172 0 86 0;
570 #X connect 173 0 128 0;

```

```

#N canvas 0 0 1265 702 10;
#X obj 350 634 rx-export~;
#X obj 593 554 pack f f f f;
#X obj 590 526 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
5 1;
#X obj 617 526 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 1
1;
#X obj 638 522 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
1;
10 #X obj 662 523 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
1;
#X obj 532 451 vline~;
#X obj 394 329 sel 0 8;
#X obj 383 12 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 1
15 1;
#X obj 352 420 vline~;
#X obj 470 416 vline~;
#X obj 351 465 max~ 0;
#X obj 408 484 max~ 0;
20 #X obj 410 463 *~ -1;
#X obj 468 346 sel 4 12;
#X obj 466 442 *~;
#X obj 590 481 unpack f f f f;
#X msg 591 430 0 1 0 0;
25 #X obj 45 686 dac~;
#X obj 226 282 rx-kick~;
#X obj 242 241 mod 4;
#X obj 242 260 sel 2;
#X obj 313 243 mod 8;
30 #X msg 311 286 100;
#X obj 312 312 rx-shaker~;
#X obj 58 540 -~ 0.25;
#X obj 58 563 cos~;
#X obj 56 636 *~ 0.25;
35 #X obj 57 516 *~;
#X obj 115 514 /~;
#X obj 466 308 mod 16;
#X obj 393 283 mod 16;
#X obj 545 261 sel 2;
40 #X msg 541 283 128;
#X obj 533 471 lop~ 5;
#X msg 391 355 1;
#X msg 421 355 -1;
#X obj 129 428 *~;
45 #X obj 129 394 cos~;
#X obj 129 371 -~ 0.25;
#X obj 15 422 *~;
#X obj 15 390 cos~;
#X obj 15 367 -~ 0.25;
50 #X obj 311 265 sel 6;
#X obj 349 560 *~ 1;
#X obj 395 559 *~ 1;
#X obj 115 491 sig~ 8;
#X obj 15 241 phasor~ 10;
55 #X obj 16 338 *~ 4;
#X obj 65 390 cos~;
#X obj 65 367 -~ 0.25;

```

```

#X obj 179 394 cos~;
#X obj 179 371 -~ 0.25;
60 #X obj 131 341 *~ 5;
#X obj 180 417 *~ 0.25;
#X obj 63 413 *~ 0.25;
#X obj 381 41 metro 19;
#X obj 407 134 change;
65 #X obj 407 112 div 6;
#X obj 382 67 f 0;
#X obj 409 68 + 1;
#X obj 393 385 pack f 456;
#X msg 471 385 -1 \, 1 456;
70 #X obj 544 303 * 114;
#X obj 131 40 mod 96;
#X obj 131 84 sel 0 12 24 36 48 64 80;
#X obj 139 178 vline~;
#X obj 136 288 lop~ 10;
75 #X msg 130 114 12;
#X msg 208 111 16;
#X obj 136 136 * 19;
#X obj 139 203 *~;
#X obj 61 595 *~;
80 #X msg 138 156 1.4 \, 0 \$1;
#X obj 176 340 *~ 15;
#X obj 65 339 *~ 12;
#X obj 721 257 -~ 0.25;
#X obj 720 286 cos~;
85 #X obj 721 178 phasor~ 60;
#X obj 722 229 *~;
#X obj 718 313 *~;
#X obj 842 207 vline~;
#X obj 837 246 *~;
90 #X obj 836 272 lop~ 50;
#X obj 718 375 cos~;
#X obj 719 347 -~ 0.25;
#X obj 737 204 samphold~;
#X obj 734 148 vline~;
95 #X msg 843 181 1 \, 0 114;
#X obj 737 45 mod 96;
#X obj 737 67 sel 0 12 24 36 48 64 80;
#X obj 738 100 bang;
#X obj 735 127 f;
100 #X obj 602 106 div 16;
#X obj 602 127 change;
#X obj 600 148 mod 4;
#X obj 600 174 sel 0 1 2 3;
#X msg 628 200 9;
105 #X msg 650 197 7;
#X msg 670 199 8;
#X msg 602 197 6;
#X obj 382 85 t f f;
#X obj 737 16 - 24;
110 #X msg 547 325 8 \, 1.5 \$1;
#X obj 545 193 mod 64;
#X obj 719 450 *~;
#X obj 750 451 *~;
#X obj 765 397 sig~ 1;

```

```
115 #X obj 766 422 -~;
    #X obj 826 396 osc~ 1;
    #X obj 824 446 +~ 0.5;
    #X obj 116 636 *~ 0.25;
    #X obj 166 636 *~ 0.25;
120 #X obj 823 421 *~ 0.25;
    #X obj 466 462 +~ 0.1;
    #X connect 1 0 0 4;
    #X connect 2 0 1 0;
    #X connect 3 0 1 1;
125 #X connect 4 0 1 2;
    #X connect 5 0 1 3;
    #X connect 6 0 34 0;
    #X connect 7 0 35 0;
    #X connect 7 1 36 0;
130 #X connect 8 0 56 0;
    #X connect 9 0 11 0;
    #X connect 9 0 13 0;
    #X connect 10 0 15 0;
    #X connect 10 0 15 1;
135 #X connect 11 0 44 0;
    #X connect 12 0 45 0;
    #X connect 13 0 12 0;
    #X connect 14 0 62 0;
    #X connect 14 1 62 0;
140 #X connect 15 0 114 0;
    #X connect 16 0 2 0;
    #X connect 16 1 3 0;
    #X connect 16 2 4 0;
    #X connect 16 3 5 0;
145 #X connect 17 0 16 0;
    #X connect 19 1 27 0;
    #X connect 20 0 21 0;
    #X connect 21 0 19 0;
    #X connect 22 0 43 0;
150 #X connect 23 0 24 0;
    #X connect 24 1 27 0;
    #X connect 25 0 26 0;
    #X connect 26 0 72 0;
    #X connect 27 0 18 0;
155 #X connect 27 0 18 1;
    #X connect 28 0 25 0;
    #X connect 29 0 28 1;
    #X connect 30 0 14 0;
    #X connect 31 0 7 0;
160 #X connect 32 0 33 0;
    #X connect 33 0 63 0;
    #X connect 34 0 29 1;
    #X connect 34 0 0 3;
    #X connect 35 0 61 0;
165 #X connect 36 0 61 0;
    #X connect 37 0 28 0;
    #X connect 38 0 37 0;
    #X connect 39 0 38 0;
    #X connect 40 0 28 0;
170 #X connect 41 0 40 0;
    #X connect 42 0 41 0;
```

```
#X connect 43 0 23 0;
#X connect 44 0 0 0;
#X connect 44 0 40 1;
175 #X connect 45 0 0 2;
#X connect 45 0 37 1;
#X connect 46 0 29 0;
#X connect 47 0 48 0;
#X connect 47 0 53 0;
180 #X connect 47 0 74 0;
#X connect 47 0 75 0;
#X connect 48 0 42 0;
#X connect 49 0 55 0;
#X connect 50 0 49 0;
185 #X connect 51 0 54 0;
#X connect 52 0 51 0;
#X connect 53 0 39 0;
#X connect 54 0 39 0;
#X connect 55 0 42 0;
190 #X connect 56 0 59 0;
#X connect 57 0 20 0;
#X connect 57 0 22 0;
#X connect 57 0 31 0;
#X connect 57 0 30 0;
195 #X connect 57 0 93 0;
#X connect 57 0 104 0;
#X connect 58 0 57 0;
#X connect 59 0 60 0;
#X connect 59 0 64 0;
200 #X connect 59 0 101 0;
#X connect 60 0 59 1;
#X connect 61 0 9 0;
#X connect 62 0 10 0;
#X connect 63 0 103 0;
205 #X connect 64 0 65 0;
#X connect 65 0 68 0;
#X connect 65 1 68 0;
#X connect 65 2 68 0;
#X connect 65 3 68 0;
210 #X connect 65 4 69 0;
#X connect 65 5 69 0;
#X connect 65 6 69 0;
#X connect 66 0 71 0;
#X connect 66 0 71 1;
215 #X connect 67 0 72 1;
#X connect 68 0 70 0;
#X connect 69 0 70 0;
#X connect 70 0 73 0;
#X connect 71 0 67 0;
220 #X connect 72 0 27 0;
#X connect 73 0 66 0;
#X connect 74 0 52 0;
#X connect 75 0 50 0;
#X connect 76 0 77 0;
225 #X connect 77 0 80 0;
#X connect 78 0 79 0;
#X connect 78 0 86 1;
#X connect 79 0 76 0;
```

```

#X connect 80 0 85 0;
230 #X connect 81 0 82 0;
#X connect 81 0 82 1;
#X connect 82 0 83 0;
#X connect 83 0 80 1;
#X connect 84 0 105 0;
235 #X connect 84 0 106 0;
#X connect 85 0 84 0;
#X connect 86 0 79 1;
#X connect 87 0 86 0;
#X connect 88 0 81 0;
240 #X connect 89 0 90 0;
#X connect 90 0 91 0;
#X connect 90 1 91 0;
#X connect 90 2 91 0;
#X connect 90 3 91 0;
245 #X connect 90 4 91 0;
#X connect 90 5 91 0;
#X connect 90 6 91 0;
#X connect 91 0 88 0;
#X connect 91 0 92 0;
250 #X connect 92 0 87 0;
#X connect 93 0 94 0;
#X connect 94 0 95 0;
#X connect 95 0 96 0;
#X connect 96 0 100 0;
255 #X connect 96 1 97 0;
#X connect 96 2 98 0;
#X connect 96 3 99 0;
#X connect 97 0 92 1;
#X connect 98 0 92 1;
260 #X connect 99 0 92 1;
#X connect 100 0 92 1;
#X connect 101 0 102 0;
#X connect 101 1 58 0;
#X connect 102 0 89 0;
265 #X connect 103 0 6 0;
#X connect 104 0 32 0;
#X connect 105 0 111 0;
#X connect 106 0 112 0;
#X connect 107 0 108 0;
270 #X connect 108 0 106 1;
#X connect 109 0 113 0;
#X connect 110 0 108 1;
#X connect 110 0 105 1;
#X connect 111 0 18 0;
275 #X connect 112 0 18 1;
#X connect 113 0 110 0;
#X connect 114 0 0 1;

```

### 30 -/reflex.sh

```

#!/bin/bash
DISPLAY=":0.0" pd -stderr -nrt -r 48000 -oss -channels 2 -open reflex.pd 2>&1 |
grep "^reflex: " |
sed "s/^reflex: //g" |
5 DISPLAY=":0.1" ./reflex |

```

```
ppmtoy4m -S 444 -F 25:1 |
y4mscaler -O preset=dvd -O yscale=1/1 |
mpeg2enc -f 8 -q 2 -b 8000 -o reflex.m2v
```

### 31 -/rx-export~.pd

```
#N canvas 70 158 268 283 10;
#X obj 29 241 print reflex;
#X obj 10 68 t b b;
#X obj 61 114 snapshot~;
5 #X obj 10 44 metro 10;
#X obj 29 215 pack f f f f f 1 1 1 1;
#X obj 77 133 snapshot~;
#X obj 93 154 snapshot~;
#X obj 45 95 snapshot~;
10 #X obj 109 172 unpack f f f f;
#X obj 124 48 inlet~;
#X obj 134 68 inlet~;
#X obj 144 88 inlet~;
#X obj 154 108 inlet~;
15 #X obj 164 128 inlet;
#X obj 10 16 loadbang;
#X msg 10 90 6;
#X connect 1 0 15 0;
#X connect 1 1 2 0;
20 #X connect 1 1 5 0;
#X connect 1 1 6 0;
#X connect 1 1 7 0;
#X connect 2 0 4 2;
#X connect 3 0 1 0;
25 #X connect 4 0 0 0;
#X connect 5 0 4 3;
#X connect 6 0 4 4;
#X connect 7 0 4 1;
#X connect 8 0 4 5;
30 #X connect 8 1 4 6;
#X connect 8 2 4 7;
#X connect 8 3 4 8;
#X connect 9 0 7 0;
#X connect 10 0 2 0;
35 #X connect 11 0 5 0;
#X connect 12 0 6 0;
#X connect 13 0 8 0;
#X connect 14 0 3 0;
#X connect 15 0 4 0;
```

### 32 -/rx-kick~.pd

```
#N canvas 674 38 450 375 10;
#X obj 60 13 inlet;
#X obj 60 282 outlet~;
#X obj 60 100 *~;
5 #X obj 44 74 vline~;
#X obj 61 120 *~;
#X obj 60 140 *~;
#X obj 59 166 *~ 30;
```

```

#X obj 59 190 -~ 0.25;
10 #X obj 61 217 cos~;
#X obj 61 238 *~;
#X obj 59 260 expr~ tanh($v1);
#X obj 9 283 outlet~;
#X msg 59 43 1 \, 0 500;
15 #X obj 124 172 *~ 4;
#X connect 0 0 12 0;
#X connect 2 0 4 0;
#X connect 2 0 4 1;
#X connect 3 0 2 0;
20 #X connect 3 0 2 1;
#X connect 3 0 11 0;
#X connect 4 0 5 0;
#X connect 4 0 5 1;
#X connect 4 0 13 0;
25 #X connect 5 0 6 0;
#X connect 6 0 7 0;
#X connect 7 0 8 0;
#X connect 8 0 9 0;
#X connect 9 0 10 0;
30 #X connect 10 0 1 0;
#X connect 12 0 3 0;
#X connect 13 0 9 1;

```

### 33 -/rx-move.pd

```

#N canvas 0 0 450 425 10;
#X obj 110 109 vsl 30 110 -5 5 0 0 \ $0-x-s \ $0-x-r empty 0 -9 0 10
-162280 -1 -1 5950 1;
#X obj 150 109 vsl 30 110 -5 5 0 0 \ $0-y-s \ $0-y-r empty 0 -9 0 10
5 -128992 -1 -1 5950 1;
#X obj 190 109 vsl 30 110 -5 5 0 0 \ $0-z-s \ $0-z-r empty 0 -9 0 10
-99865 -1 -1 5950 1;
#X obj 39 33 loadbang;
#X msg 39 54 set 0;
10 #X obj 105 257 r \ $0-x-s;
#X obj 165 257 r \ $0-y-s;
#X obj 226 258 r \ $0-z-s;
#X obj 105 311 pack 0 0 0;
#X obj 105 377 outlet;
15 #X msg 105 343 Move \ $1 \ $2 \ $3;
#X obj 113 284 t b f;
#X obj 163 284 t b f;
#X connect 3 0 4 0;
#X connect 4 0 0 0;
20 #X connect 4 0 1 0;
#X connect 4 0 2 0;
#X connect 5 0 8 0;
#X connect 6 0 11 0;
#X connect 7 0 12 0;
25 #X connect 8 0 10 0;
#X connect 10 0 9 0;
#X connect 11 0 8 0;
#X connect 11 1 8 1;
#X connect 12 0 8 0;
30 #X connect 12 1 8 2;

```

```
#X coords 0 -1 1 1 128 128 2 100 100;
```

### 34 -/rx-ology.pd

```
#N canvas 70 158 450 469 10;
#X obj 14 23 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 1 1
;
#X msg 35 10 \; pd dsp 1;
5 #X obj 9 356 print reflex;
#X obj 30 71 loadbang;
#X msg 30 95 1;
#X obj 12 135 t b b;
#X obj 42 229 snapshot~;
10 #X obj 219 335 dac~;
#X obj 12 44 metro 10;
#X msg 10 160 6;
#X obj 138 300 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 1
1;
15 #X obj 121 299 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 1
1;
#X obj 102 297 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 1
1;
#X obj 82 296 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 1
20 1;
#X obj 8 328 pack f f f f f 1 1 1 1;
#X obj 55 248 snapshot~;
#X obj 71 267 snapshot~;
#X obj 25 192 catch~ \ $0-red;
25 #X connect 0 0 8 0;
#X connect 3 0 4 0;
#X connect 5 0 9 0;
#X connect 5 1 6 0;
#X connect 5 1 15 0;
30 #X connect 5 1 16 0;
#X connect 6 0 14 2;
#X connect 8 0 5 0;
#X connect 9 0 14 0;
#X connect 10 0 14 8;
35 #X connect 11 0 14 7;
#X connect 12 0 14 6;
#X connect 13 0 14 5;
#X connect 14 0 2 0;
#X connect 15 0 14 3;
40 #X connect 16 0 14 4;
```

### 35 -/rx-pack8.pd

```
#N canvas 0 0 450 300 10;
#X obj 20 23 inlet;
#X obj 60 23 inlet;
#X obj 100 23 inlet;
5 #X obj 140 23 inlet;
#X obj 180 23 inlet;
#X obj 220 23 inlet;
#X obj 260 23 inlet;
#X obj 300 23 inlet;
```

```

10  #X obj 60 43 t b f;
    #X obj 100 43 t b f;
    #X obj 140 43 t b f;
    #X obj 180 43 t b f;
    #X obj 220 43 t b f;
15  #X obj 260 43 t b f;
    #X obj 300 43 t b f;
    #X obj 110 137 pack f f f f f f f f;
    #X obj 110 166 outlet;
    #X connect 0 0 15 0;
20  #X connect 1 0 8 0;
    #X connect 2 0 9 0;
    #X connect 3 0 10 0;
    #X connect 4 0 11 0;
    #X connect 5 0 12 0;
25  #X connect 6 0 13 0;
    #X connect 7 0 14 0;
    #X connect 8 0 15 0;
    #X connect 8 1 15 1;
    #X connect 9 0 15 0;
30  #X connect 9 1 15 2;
    #X connect 10 0 15 0;
    #X connect 10 1 15 3;
    #X connect 11 0 15 0;
    #X connect 11 1 15 4;
35  #X connect 12 0 15 0;
    #X connect 12 1 15 5;
    #X connect 13 0 15 0;
    #X connect 13 1 15 6;
    #X connect 14 0 15 0;
40  #X connect 14 1 15 7;
    #X connect 15 0 16 0;

```

### 36 -/rx-shaker~.pd

```

    #N canvas 687 161 450 300 10;
    #X obj 28 22 inlet;
    #X obj 85 265 outlet~;
    #X obj 28 103 cos~;
5  #X obj 27 82 vline~;
    #X obj 28 134 +~ 1;
    #X obj 28 161 *~ 0.5;
    #X obj 91 222 *~;
    #X obj 118 136 noise~;
10 #X obj 117 163 samphold~;
    #X obj 171 138 phasor~ 6000;
    #X obj 24 265 outlet~;
    #X msg 27 54 -0.5 \, 0.5 \ $1;
    #X connect 0 0 11 0;
15 #X connect 2 0 4 0;
    #X connect 3 0 2 0;
    #X connect 4 0 5 0;
    #X connect 5 0 6 0;
    #X connect 5 0 10 0;
20 #X connect 6 0 1 0;
    #X connect 7 0 8 0;
    #X connect 8 0 6 1;

```

```
#X connect 9 0 8 1;
#X connect 11 0 3 0;
```

### 37 -/rx-shape.pd

```
#N canvas 499 72 450 469 10;
#X obj 106 116 vsl 23 96 0 1 0 0 \ $0-x-s \ $0-x-r empty 0 -9 0 10 -191407
-1 -1 1100 1;
#X obj 136 116 vsl 23 96 0 1 0 0 \ $0-y-s \ $0-y-r empty 0 -9 0 10 -162280
5 -1 -1 900 1;
#X obj 166 116 vsl 23 96 0 1 0 0 \ $0-z-s \ $0-z-r empty 0 -9 0 10 -128992
-1 -1 1700 1;
#X obj 196 116 vsl 23 96 0 1 0 0 \ $0-w-s \ $0-w-r empty 0 -9 0 10 -99865
-1 -1 2600 1;
10 #X obj 105 35 loadbang;
#X msg 104 58 set 0.5;
#X obj 101 320 pack 0.5 0.5 0.5 0.5;
#X obj 66 255 r \ $0-x-s;
#X obj 126 255 r \ $0-y-s;
15 #X obj 187 256 r \ $0-z-s;
#X obj 113 284 t b f;
#X obj 163 284 t b f;
#X obj 233 285 t b f;
#X obj 257 257 r \ $0-w-s;
20 #X msg 101 356 Shape \ $1 \ $2 \ $3 \ $4;
#X obj 103 391 outlet;
#X obj 106 105 cnv 5 6 6 empty empty empty 20 12 0 14 -258113 -66577
0;
#X obj 114 105 cnv 5 6 6 empty empty empty 20 12 0 14 -257985 -66577
25 0;
#X obj 122 105 cnv 5 6 6 empty empty empty 20 12 0 14 -4034 -66577
0;
#X obj 136 105 cnv 5 6 6 empty empty empty 20 12 0 14 -258113 -66577
0;
30 #X obj 144 105 cnv 5 6 6 empty empty empty 20 12 0 14 -4032 -66577
0;
#X obj 152 105 cnv 5 6 6 empty empty empty 20 12 0 14 -4160 -66577
0;
#X obj 166 105 cnv 5 6 6 empty empty empty 20 12 0 14 -257985 -66577
35 0;
#X obj 174 105 cnv 5 6 6 empty empty empty 20 12 0 14 -4032 -66577
0;
#X obj 182 105 cnv 5 6 6 empty empty empty 20 12 0 14 -159808 -66577
0;
40 #X obj 196 105 cnv 5 6 6 empty empty empty 20 12 0 14 -4034 -66577
0;
#X obj 204 105 cnv 5 6 6 empty empty empty 20 12 0 14 -4160 -66577
0;
#X obj 212 105 cnv 5 6 6 empty empty empty 20 12 0 14 -159808 -66577
45 0;
#X obj 106 218 cnv 5 6 6 empty empty empty 20 12 0 14 -4032 -66577
0;
#X obj 114 218 cnv 5 6 6 empty empty empty 20 12 0 14 -4160 -66577
0;
50 #X obj 122 218 cnv 5 6 6 empty empty empty 20 12 0 14 -159808 -66577
0;
#X obj 152 218 cnv 5 6 6 empty empty empty 20 12 0 14 -159808 -66577
```

```

0;
#X obj 182 218 cnv 5 6 6 empty empty empty 20 12 0 14 -4160 -66577
55 0;
#X obj 213 218 cnv 5 6 6 empty empty empty 20 12 0 14 -4032 -66577
0;
#X obj 135 218 cnv 5 6 6 empty empty empty 20 12 0 14 -257985 -66577
0;
60 #X obj 144 218 cnv 5 6 6 empty empty empty 20 12 0 14 -4034 -66577
0;
#X obj 165 218 cnv 5 6 6 empty empty empty 20 12 0 14 -258113 -66577
0;
#X obj 174 218 cnv 5 6 6 empty empty empty 20 12 0 14 -4034 -66577
65 0;
#X obj 196 218 cnv 5 6 6 empty empty empty 20 12 0 14 -258113 -66577
0;
#X obj 204 218 cnv 5 6 6 empty empty empty 20 12 0 14 -257985 -66577
0;
70 #X connect 4 0 5 0;
#X connect 5 0 0 0;
#X connect 5 0 1 0;
#X connect 5 0 2 0;
#X connect 5 0 3 0;
75 #X connect 6 0 14 0;
#X connect 7 0 6 0;
#X connect 8 0 10 0;
#X connect 9 0 11 0;
#X connect 10 0 6 0;
80 #X connect 10 1 6 1;
#X connect 11 0 6 0;
#X connect 11 1 6 2;
#X connect 12 0 6 0;
#X connect 12 1 6 3;
85 #X connect 13 0 12 0;
#X connect 14 0 15 0;
#X coords 0 -1 1 1 128 128 2 100 100;

```

### 38 -/rx-solid.pd

```

#N canvas 0 0 450 500 10;
#X obj 109 147 hradio 28 1 0 4 \ $0-axis-s \ $0-axis-r empty 0 -8 0 10
-162280 -1 -1 3;
#X obj 117 241 r \ $0-axis-s;
5 #X obj 117 270 sel 0 1 2 3;
#X obj 117 389 outlet;
#X msg 117 291 XY 0 \, XZ 0 \, XW 0 \, YZ 1 \, YW 1 \, ZW 1;
#X msg 132 312 XY 0 \, XZ 1 \, XW 1 \, YZ 0 \, YW 0 \, ZW 1;
#X msg 148 334 XY 1 \, XZ 0 \, XW 1 \, YZ 0 \, YW 1 \, ZW 0;
10 #X msg 165 355 XY 1 \, XZ 1 \, XW 0 \, YZ 1 \, YW 0 \, ZW 0;
#X connect 1 0 2 0;
#X connect 2 0 4 0;
#X connect 2 1 5 0;
#X connect 2 2 6 0;
15 #X connect 2 3 7 0;
#X connect 4 0 3 0;
#X connect 5 0 3 0;
#X connect 6 0 3 0;
#X connect 7 0 3 0;

```

```
20 #X coords 0 -1 1 1 128 128 2 100 100;
```

### 39 \_/rx-visibility.pd

```
#N canvas 0 0 640 471 10;
#X obj 109 109 tgl 30 0 \ $0-xy-s \ $0-xy-r empty 17 7 0 10 -258113 -1
-1 0 1;
#X obj 149 109 tgl 30 0 \ $0-xz-s \ $0-xz-r empty 17 7 0 10 -257985 -1
5 -1 0 1;
#X obj 189 109 tgl 30 0 \ $0-xw-s \ $0-xw-r empty 17 7 0 10 -4034 -1
-1 0 1;
#X obj 189 149 tgl 30 0 \ $0-yw-s \ $0-yw-r empty 17 7 0 10 -4160 -1
-1 0 1;
10 #X obj 149 149 tgl 30 0 \ $0-yz-s \ $0-yz-r empty 17 7 0 10 -4032 -1
-1 0 1;
#X obj 189 189 tgl 30 0 \ $0-zw-s \ $0-zw-r empty 17 7 0 10 -159808 -1
-1 0 1;
#X obj 291 381 outlet;
15 #X obj 47 252 r \ $0-xy-s;
#X obj 46 275 sel 0 1;
#X msg 25 316 XY Off;
#X msg 74 316 XY On;
#X obj 136 275 sel 0 1;
20 #X obj 226 275 sel 0 1;
#X obj 316 275 sel 0 1;
#X obj 406 275 sel 0 1;
#X obj 496 275 sel 0 1;
#X obj 137 252 r \ $0-xz-s;
25 #X msg 115 316 XZ Off;
#X msg 164 316 XZ On;
#X obj 227 252 r \ $0-xw-s;
#X msg 205 316 XW Off;
#X msg 254 316 XW On;
30 #X obj 317 252 r \ $0-yz-s;
#X msg 295 316 YZ Off;
#X msg 344 316 YZ On;
#X msg 385 316 YW Off;
#X msg 434 316 YW On;
35 #X obj 406 251 r \ $0-yw-s;
#X obj 497 252 r \ $0-zw-s;
#X msg 475 316 ZW Off;
#X msg 524 316 ZW On;
#X obj 295 32 inlet;
40 #X obj 295 57 route XY XZ XW YZ YW ZW;
#X obj 296 201 s \ $0-xy-r;
#X obj 316 182 s \ $0-xz-r;
#X obj 340 163 s \ $0-xw-r;
#X obj 362 145 s \ $0-yz-r;
45 #X obj 385 125 s \ $0-yw-r;
#X obj 407 104 s \ $0-zw-r;
#X connect 7 0 8 0;
#X connect 8 0 9 0;
#X connect 8 1 10 0;
50 #X connect 9 0 6 0;
#X connect 10 0 6 0;
#X connect 11 0 17 0;
#X connect 11 1 18 0;
```

```

#X connect 12 0 20 0;
55 #X connect 12 1 21 0;
#X connect 13 0 23 0;
#X connect 13 1 24 0;
#X connect 14 0 25 0;
#X connect 14 1 26 0;
60 #X connect 15 0 29 0;
#X connect 15 1 30 0;
#X connect 16 0 11 0;
#X connect 17 0 6 0;
#X connect 18 0 6 0;
65 #X connect 19 0 12 0;
#X connect 20 0 6 0;
#X connect 21 0 6 0;
#X connect 22 0 13 0;
#X connect 23 0 6 0;
70 #X connect 24 0 6 0;
#X connect 25 0 6 0;
#X connect 26 0 6 0;
#X connect 27 0 14 0;
#X connect 28 0 15 0;
75 #X connect 29 0 6 0;
#X connect 30 0 6 0;
#X connect 31 0 32 0;
#X connect 32 0 33 0;
#X connect 32 1 34 0;
80 #X connect 32 2 35 0;
#X connect 32 3 36 0;
#X connect 32 4 37 0;
#X connect 32 5 38 0;
#X coords 0 -1 1 1 128 128 2 100 100;

```

## 40 -/Snapshot.hs

```

{-

fl4m6e -- fast fractal flame generator
Copyright (C) 2009 Claude Heiland-Allen <claude@mathr.co.uk>

5 This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

10 This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU Affero General Public License for more details.

15 You should have received a copy of the GNU Affero General Public License
along with this program. If not, see <http://www.gnu.org/licenses/>.

-}

20 {-

save a snapshot of OpenGL's front display buffer to a handle

```

```

25  -}

module Snapshot (hSnapshot, writeSnapshot, snapshotWith) where

import Control.Monad(forM_)
30  import System.IO(Handle())
import Graphics.UI.GLUT(
    readPixels,
    Position,
    Size(Size),
35  PixelData(PixelData),
    PixelFormat(OpenGL),
    DataType(UnsignedByte))
import Foreign.Marshal.Alloc(allocaBytes)
import Foreign.Ptr(plusPtr)
40  import qualified Data.ByteString.Internal as BSI
import qualified Data.ByteString as BS

-- save a screenshot to a handle as binary PPM
snapshotWith :: (BS.ByteString -> IO b) -> Position -> Size -> IO b
45  snapshotWith f p0 vp@(Size vw vh) = do
    let fi q = fromIntegral q
        p6 = "P6\n" ++ show vw ++ " " ++ show vh ++ " 255\n"
    allocaBytes (fi (vw*vh*3)) $ \ptr -> do
        readPixels p0 vp $ PixelData OpenGL UnsignedByte ptr
50  px <- BSI.create (fi $ vw * vh * 3) $ \d -> forM_ [0..vh-1] $ \y ->
        BSI.memcpy
            (d `plusPtr` fi (y*vw*3))
            (ptr `plusPtr` fi ((vh-1-y)*vw*3))
            (fi (vw*3))
55  f $ BS.pack (map (toEnum . fromEnum) p6) 'BS.append' px

hSnapshot :: Handle -> Position -> Size -> IO ()
hSnapshot h = snapshotWith (BS.hPutStr h)

60  writeSnapshot :: FilePath -> Position -> Size -> IO ()
writeSnapshot f = snapshotWith (BS.writeFile f)

```

## 41 Snapshot.hs

```

{-
fl4m6e -- fast fractal flame generator
Copyright (C) 2009 Claude Heiland-Allen <claude@mathr.co.uk>
5
This program is free software: you can redistribute it and/or modify
it under the terms of the GNU Affero General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.
10
This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU Affero General Public License for more details.
15
You should have received a copy of the GNU Affero General Public License

```

along with this program. If not, see <<http://www.gnu.org/licenses/>>.

```

-}
20 {-

save a snapshot of OpenGL's front display buffer to a handle

25 -}

module Snapshot (hSnapshot, writeSnapshot, snapshotWith) where

import Control.Monad(forM_)
30 import System.IO(Handle())
import Graphics.UI.GLUT(
    ($=),
    rowAlignment,
    PixelStoreDirection(..),
35    readPixels,
    Position,
    Size(Size),
    PixelData(PixelData),
    PixelFormat(OpenGL),
    DataType(UnsignedByte))
40 import Foreign.Marshal.Alloc(allocaBytes)
import Foreign.Ptr(plusPtr)
import qualified Data.ByteString.Internal as BSI
import qualified Data.ByteString as BS

45 -- save a screenshot to a handle as binary PPM
snapshotWith :: (BS.ByteString -> IO b) -> Position -> Size -> IO b
snapshotWith f p0 vp@(Size vw vh) = do
    let fi q = fromIntegral q
50    p6 = "P6\n" ++ show vw ++ " " ++ show vh ++ " 255\n"
    allocaBytes (fi (vw*vh*3)) $ \ptr -> do
        rowAlignment Pack $= 1
        rowAlignment Unpack $= 1
        readPixels p0 vp $ PixelData RGB UnsignedByte ptr
55    px <- BSI.create (fi $ vw * vh * 3) $ \d -> forM_ [0..vh-1] $ \y ->
        BSI.memcpy
            (d `plusPtr` fi (y*vw*3))
            (ptr `plusPtr` fi ((vh-1-y)*vw*3))
            (fi (vw*3))
60    f $ BS.pack (map (toEnum . fromEnum) p6) `BS.append` px

hSnapshot :: Handle -> Position -> Size -> IO ()
hSnapshot h = snapshotWith (BS.hPutStr h)

65 writeSnapshot :: FilePath -> Position -> Size -> IO ()
writeSnapshot f = snapshotWith (BS.writeFile f)

```

## 42 snare~.pd

```

#N canvas 0 0 345 478 10;
#X obj 112 457 outlet~;
#X obj 39 8 inlet;
#X obj 146 94 noise~;

```

```
5  #X obj 34 178 osc~;
   #X obj 100 300 *~;
   #X obj 128 167 *~;
   #X obj 113 324 *~;
   #X obj 170 292 *~;
10  #X obj 14 100 *~;
   #X obj 68 322 *~;
   #X obj 14 125 *~ 12;
   #X obj 107 354 clip~ 0 1;
   #X obj 34 201 clip~ -0.8 0.8;
15  #X obj 166 316 *~ 0.1;
   #X obj 108 375 *~ 0.4;
   #X obj 54 359 *~ 0.9;
   #X obj 146 118 bp~ 3300 0.5;
   #X obj 146 142 lop~ 7000;
20  #X obj 112 428 bp~ 8000 1;
   #X obj 73 263 clip~ -0.9 0.9;
   #X obj 71 241 *~ 10000;
   #X obj 15 154 +~ 150;
   #X obj 113 399 *~ 0.35;
25  #X obj 40 70 cm-ead~ 1 300;
   #X obj 38 49 t b f;
   #X obj 116 50 * 10;
   #X obj 37 29 sel 0;
   #X connect 1 0 26 0;
30  #X connect 2 0 16 0;
   #X connect 3 0 12 0;
   #X connect 4 0 6 0;
   #X connect 5 0 6 1;
   #X connect 6 0 11 0;
35  #X connect 7 0 13 0;
   #X connect 8 0 9 1;
   #X connect 8 0 10 0;
   #X connect 9 0 15 0;
   #X connect 10 0 21 0;
40  #X connect 11 0 14 0;
   #X connect 12 0 9 0;
   #X connect 12 0 20 0;
   #X connect 13 0 22 0;
   #X connect 14 0 22 0;
45  #X connect 15 0 22 0;
   #X connect 16 0 7 0;
   #X connect 16 0 17 0;
   #X connect 17 0 5 1;
   #X connect 18 0 0 0;
50  #X connect 19 0 4 0;
   #X connect 20 0 19 0;
   #X connect 21 0 3 0;
   #X connect 22 0 18 0;
   #X connect 23 0 4 1;
55  #X connect 23 0 5 0;
   #X connect 23 0 8 0;
   #X connect 23 0 8 1;
   #X connect 23 0 7 1;
   #X connect 24 0 23 0;
60  #X connect 24 1 25 0;
   #X connect 25 0 23 2;
```

```
#X connect 26 1 24 0;
```

## 43 START.sh

```
#!/bin/bash
pd -path /usr/lib/pd/extra/zexy -lib zexy -stderr -open reflex.pd &
sleep 5
pdreceive 5555 | ./reflex 3 4 3 -geometry 1024x768 &
5 wait
```

## 44 \_/Vector.hs

```
{-# LANGUAGE MultiParamTypeClasses, TypeSynonymInstances #-}

module Vector where

5 type R = Double

data V1 = V1 !R deriving (Show, Eq, Ord)
data V2 = V2 !R !R deriving (Show, Eq, Ord)
data V3 = V3 !R !R !R deriving (Show, Eq, Ord)
10 data V4 = V4 !R !R !R !R deriving (Show, Eq, Ord)

class V a where
  o :: a
  (^+^) :: a -> a -> a
  (^-^) :: a -> a -> a
  (*^ ) :: R -> a -> a
  (^* ) :: a -> R -> a
  (^/ ) :: a -> R -> a
  dot :: a -> a -> R
20

(|-|) :: V a => a -> a -> R
u |-| v = let d = u ^-^ v in d `dot` d

norm :: V a => a -> a
25 norm v = v ^/ sqrt (v `dot` v)

instance V V1 where
  o = V1 0
  V1 a ^+^ V1 x = V1 (a + x)
  V1 a ^-^ V1 x = V1 (a - x)
30 k *^ V1 x = V1 (k * x)
  V1 a ^* k = V1 (a * k)
  V1 a ^/ k = V1 (a / k)
  V1 a `dot` V1 x = a * x
35

instance V V2 where
  o = V2 0 0
  V2 a b ^+^ V2 x y = V2 (a + x) (b + y)
  V2 a b ^-^ V2 x y = V2 (a - x) (b - y)
40 k *^ V2 x y = V2 (k * x) (k * y)
  V2 a b ^* k = V2 (a * k) (b * k)
  V2 a b ^/ k = V2 (a / k) (b / k)
  V2 a b `dot` V2 x y = a * x + b * y
```

87

```

M2 a11 a12 a21 a22 ^^+^^ M2 b11 b12 b21 b22 =
  M2 (a11 + b11) (a12 + b12) (a21 + b21) (a22 + b22)
M2 a11 a12 a21 a22 ^^--^^ M2 b11 b12 b21 b22 =
  M2 (a11 - b11) (a12 - b12) (a21 - b21) (a22 - b22)
105 M2 a11 a12 a21 a22 ^^*^^ M2 b11 b12 b21 b22 =
  M2 (a11*b11 + a12*b21) (a11*b12 + a12*b22) (a21*b11 + a22*b21) (a21*b12 +
    ↪ a22*b22)
  a *^^ M2 b11 b12 b21 b22 = M2 (a * b11) (a * b12) (a * b21) (a * b22)
M2 a11 a12 a21 a22 ^^* b = M2 (a11 * b) (a12 * b) (a21 * b) (a22 * b)
M2 a11 a12 a21 a22 ^^/ b = M2 (a11 / b) (a12 / b) (a21 / b) (a22 / b)
110 M2 a11 a12 a21 a22 'mdot' M2 b11 b12 b21 b22 =
  a11*b11 + a12*b12 + a21*b21 + a22*b22
det (M2 a11 a12 a21 a22) = a11 * a22 - a12 * a21
inv a@(M2 a11 a12 a21 a22) = (M2 a22 (-a12) (-a21) a11) ^^/ det a

115 instance M M3 where
  i = M3 1 0 0 0 1 0 0 0 1
  M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 ^^+^^ M3 b11 b12 b13 b21 b22 b23 b31 ↪
    ↪ b32 b33 =
    M3 (a11 + b11) (a12 + b12) (a13 + b13) (a21 + b21) (a22 + b22) (a23 + b23) (↪
      ↪ a31 + b31) (a32 + b32) (a33 + b33)
  M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 ^^--^^ M3 b11 b12 b13 b21 b22 b23 b31 ↪
    ↪ b32 b33 =
120 M3 (a11 - b11) (a12 - b12) (a13 - b13) (a21 - b21) (a22 - b22) (a23 - b23) (↪
  ↪ a31 - b31) (a32 - b32) (a33 - b33)
  M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 ^^*^^ M3 b11 b12 b13 b21 b22 b23 b31 ↪
    ↪ b32 b33 =
    M3 (a11*b11 + a12*b21 + a13*b31) (a11*b12 + a12*b22 + a13*b32) (a11*b13 + ↪
      ↪ a12*b23 + a13*b33)
      (a21*b11 + a22*b21 + a23*b31) (a21*b12 + a22*b22 + a23*b32) (a21*b13 + ↪
        ↪ a22*b23 + a23*b33)
        (a31*b11 + a32*b21 + a33*b31) (a31*b12 + a32*b22 + a33*b32) (a31*b13 + ↪
          ↪ a32*b23 + a33*b33)
125 a *^^ M3 b11 b12 b13 b21 b22 b23 b31 b32 b33 =
  M3 (a * b11) (a * b12) (a * b13) (a * b21) (a * b22) (a * b23) (a * b31) (a ↪
    ↪ * b32) (a * b33)
  M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 ^^* b =
  M3 (a11 * b) (a12 * b) (a13 * b) (a21 * b) (a22 * b) (a23 * b) (a31 * b) (↪
    ↪ a32 * b) (a33 * b)
  M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 ^^/ b =
130 M3 (a11 / b) (a12 / b) (a13 / b) (a21 / b) (a22 / b) (a23 / b) (a31 / b) (↪
  ↪ a32 / b) (a33 / b)
  M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 'mdot' M3 b11 b12 b13 b21 b22 b23 b31 ↪
    ↪ b32 b33 =
    a11*b11 + a12*b12 + a13*b13 +
    a21*b21 + a22*b22 + a23*b23 +
    a31*b31 + a32*b32 + a33*b33
135 det (M3 a11 a12 a13 a21 a22 a23 a31 a32 a33) =
  let m11 = M2 a22 a23 a32 a33
    m12 = M2 a21 a23 a31 a33
    m13 = M2 a21 a22 a31 a32
  in a11 * det m11 - a12 * det m12 + a13 * det m13
140 inv a@(M3 a11 a12 a13 a21 a22 a23 a31 a32 a33) =
  let m11 = a33 * a22 - a32 * a23
    m12 = -(a33 * a12 - a32 * a13)
    m13 = a23 * a12 - a22 * a13
    m21 = -(a33 * a21 - a31 * a23)

```

```

145      m22 = a33 * a11 - a31 * a13
      m23 = -(a23 * a11 - a21 * a13)
      m31 = a32 * a21 - a31 * a22
      m32 = -(a32 * a11 - a31 * a12)
      m33 = a22 * a11 - a21 * a12
150      in (M3 m11 m12 m13 m21 m22 m23 m31 m32 m33) ^^ / det a

instance M M4 where
  i = M4 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1
  M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 ^^+^^ M4 ↯
    ↪ b11 b12 b13 b14 b21 b22 b23 b24 b31 b32 b33 b34 b41 b42 b43 b44 =
155      M4 (a11 + b11) (a12 + b12) (a13 + b13) (a14 + b14)
        (a21 + b21) (a22 + b22) (a23 + b23) (a24 + b24)
        (a31 + b31) (a32 + b32) (a33 + b33) (a34 + b34)
        (a41 + b41) (a42 + b42) (a43 + b43) (a44 + b44)
  M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 ^^--^^ M4 ↯
    ↪ b11 b12 b13 b14 b21 b22 b23 b24 b31 b32 b33 b34 b41 b42 b43 b44 =
160      M4 (a11 - b11) (a12 - b12) (a13 - b13) (a14 - b14)
        (a21 - b21) (a22 - b22) (a23 - b23) (a24 - b24)
        (a31 - b31) (a32 - b32) (a33 - b33) (a34 - b34)
        (a41 - b41) (a42 - b42) (a43 - b43) (a44 - b44)
  M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 ^^*^^ M4 ↯
    ↪ b11 b12 b13 b14 b21 b22 b23 b24 b31 b32 b33 b34 b41 b42 b43 b44 =
165      M4 (a11*b11 + a12*b21 + a13*b31 + a14*b41) (a11*b12 + a12*b22 + a13*b32 + ↯
        ↪ a14*b42) (a11*b13 + a12*b23 + a13*b33 + a14*b43) (a11*b14 + a12*b24 + ↯
        ↪ a13*b34 + a14*b44)
        (a21*b11 + a22*b21 + a23*b31 + a24*b41) (a21*b12 + a22*b22 + a23*b32 + ↯
        ↪ a24*b42) (a21*b13 + a22*b23 + a23*b33 + a24*b43) (a21*b14 + a22*b24 ↯
        ↪ + a23*b34 + a24*b44)
        (a31*b11 + a32*b21 + a33*b31 + a34*b41) (a31*b12 + a32*b22 + a33*b32 + ↯
        ↪ a34*b42) (a31*b13 + a32*b23 + a33*b33 + a34*b43) (a31*b14 + a32*b24 ↯
        ↪ + a33*b34 + a34*b44)
        (a41*b11 + a42*b21 + a43*b31 + a44*b41) (a41*b12 + a42*b22 + a43*b32 + ↯
        ↪ a44*b42) (a41*b13 + a42*b23 + a43*b33 + a44*b43) (a41*b14 + a42*b24 ↯
        ↪ + a43*b34 + a44*b44)
  a *^^ M4 b11 b12 b13 b14 b21 b22 b23 b24 b31 b32 b33 b34 b41 b42 b43 b44 =
170      M4 (a * b11) (a * b12) (a * b13) (a * b14)
        (a * b21) (a * b22) (a * b23) (a * b24)
        (a * b31) (a * b32) (a * b33) (a * b34)
        (a * b41) (a * b42) (a * b43) (a * b44)
  M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 ^^* b =
175      M4 (a11 * b) (a12 * b) (a13 * b) (a14 * b)
        (a21 * b) (a22 * b) (a23 * b) (a24 * b)
        (a31 * b) (a32 * b) (a33 * b) (a34 * b)
        (a41 * b) (a42 * b) (a43 * b) (a44 * b)
  M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 ^^/ b =
180      M4 (a11 / b) (a12 / b) (a13 / b) (a14 / b)
        (a21 / b) (a22 / b) (a23 / b) (a24 / b)
        (a31 / b) (a32 / b) (a33 / b) (a34 / b)
        (a41 / b) (a42 / b) (a43 / b) (a44 / b)
  M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 ‘mdot‘ M4 ↯
    ↪ b11 b12 b13 b14 b21 b22 b23 b24 b31 b32 b33 b34 b41 b42 b43 b44 =
185      a11*b11 + a12*b12 + a13*b13 + a14*b14 +
        a21*b21 + a22*b22 + a23*b23 + a24*b24 +
        a31*b31 + a32*b32 + a33*b33 + a34*b34 +
        a41*b41 + a42*b42 + a43*b43 + a44*b44
  det (M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44) =

```

```

190      let m11 = M3 a22 a23 a24 a32 a33 a34 a42 a43 a44
          m12 = M3 a21 a23 a24 a31 a33 a34 a41 a43 a44
          m13 = M3 a21 a22 a24 a31 a32 a34 a41 a42 a44
          m14 = M3 a21 a22 a23 a31 a32 a33 a41 a42 a43
      in a11 * det m11 - a12 * det m12 + a13 * det m13 - a14 * det m14
195  inv (M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44) =
      let a = M2 a11 a12 a21 a22
          b = M2 a13 a14 a23 a24
          c = M2 a31 a32 a41 a42
          d = M2 a33 a34 a43 a44
200      a1 = inv a
          ca1 = c ^^*^^ a1
          dcab1 = inv (d ^^-^^ (ca1 ^^*^^ b))
          a1bdcab1 = a1 ^^*^^ (b ^^*^^ dcab1)
          M2 m11 m12 m21 m22 = a1 ^^+^^ (a1bdcab1 ^^*^^ ca1)
205      M2 m13 m14 m23 m24 = (-1) ^^*^^ a1bdcab1
          M2 m31 m32 m41 m42 = (-1) ^^*^^ (dcab1 ^^*^^ ca1)
          M2 m33 m34 m43 m44 = dcab1
      in M4 m11 m12 m13 m14 m21 m22 m23 m24 m31 m32 m33 m34 m41 m42 m43 m44

210  class MV m v where
      (^^*^) :: m -> v -> v
      (^*^!) :: v -> v -> m

      reflector :: (M m, V v, MV m v) => v -> m
215  reflector v = i ^^-^^ (2 *^^ (v ^*^! v))

      instance MV M1 V1 where
          M1 m ^^*^ V1 v = V1 (m * v)
          V1 a ^*^! V1 b = M1 (a * b)
220

      instance MV M2 V2 where
          M2 m11 m12 m21 m22 ^^*^ V2 v1 v2 =
              V2 (m11*v1 + m12*v2)
                  (m21*v1 + m22*v2)
225      V2 a1 a2 ^*^! V2 b1 b2 =
          M2 (a1 * b1) (a1 * b2)
              (a2 * b1) (a2 * b2)

      instance MV M3 V3 where
230      M3 m11 m12 m13 m21 m22 m23 m31 m32 m33 ^^*^ V3 v1 v2 v3 =
          V3 (m11*v1 + m12*v2 + m13*v3)
              (m21*v1 + m22*v2 + m23*v3)
              (m31*v1 + m32*v2 + m33*v3)
          V3 a1 a2 a3 ^*^! V3 b1 b2 b3 =
235      M3 (a1 * b1) (a1 * b2) (a1 * b3)
          (a2 * b1) (a2 * b2) (a2 * b3)
          (a3 * b1) (a3 * b2) (a3 * b3)

      instance MV M4 V4 where
240      M4 m11 m12 m13 m14 m21 m22 m23 m24 m31 m32 m33 m34 m41 m42 m43 m44 ^^*^ V4 v1 v2 v3 v4 =
          V4 (m11*v1 + m12*v2 + m13*v3 + m14*v4)
              (m21*v1 + m22*v2 + m23*v3 + m24*v4)
              (m31*v1 + m32*v2 + m33*v3 + m34*v4)
              (m41*v1 + m42*v2 + m43*v3 + m44*v4)
245      V4 a1 a2 a3 a4 ^*^! V4 b1 b2 b3 b4 =

```

```

      M4 (a1 * b1) (a1 * b2) (a1 * b3) (a1 * b4)
          (a2 * b1) (a2 * b2) (a2 * b3) (a2 * b4)
          (a3 * b1) (a3 * b2) (a3 * b3) (a3 * b4)
          (a4 * b1) (a4 * b2) (a4 * b3) (a4 * b4)
250 cross4 :: V4 -> V4 -> V4 -> V4
cross4 (V4 u0 u1 u2 u3) (V4 v0 v1 v2 v3) (V4 w0 w1 w2 w3) =
{-
255   let vw01 = v0 * w1 - v1 * w0
      vw02 = v0 * w2 - v2 * w0
      vw03 = v0 * w3 - v3 * w0
      vw12 = v1 * w2 - v2 * w1
      vw13 = v1 * w3 - v3 * w1
      vw23 = v2 * w3 - v3 * w2
260   r0 = u1 * vw23 - u2 * vw13 + u3 * vw12
      r1 = - u0 * vw23 + u2 * vw03 - u3 * vw02
      r2 = u0 * vw13 - u1 * vw03 + u3 * vw02
      r3 = - u0 * vw12 + u1 * vw02 - u2 * vw01
   in V4 r0 r1 r2 r3
265 -}
   let m0 = M3 u1 u2 u3 v1 v2 v3 w1 w2 w3
       m1 = M3 u0 u2 u3 v0 v2 v3 w0 w2 w3
       m2 = M3 u0 u1 u3 v0 v1 v3 w0 w1 w3
       m3 = M3 u0 u1 u2 v0 v1 v2 w0 w1 w2
270   in V4 (det m0) (-(det m1)) (det m2) (-(det m3))

epsilon :: R
275 epsilon = 1e-3

class Approx a where
  (~=) :: a -> a -> Bool

280 instance Approx R where
  x ~= y = abs (x - y) < epsilon

instance Approx V1 where
  x ~= y = sqrt (x | - | y) ~= 0
285 instance Approx V2 where
  x ~= y = sqrt (x | - | y) ~= 0

instance Approx V3 where
290   x ~= y = sqrt (x | - | y) ~= 0

instance Approx V4 where
  x ~= y = sqrt (x | - | y) ~= 0

295 instance Approx M1 where
  x ~= y = sqrt (x || - || y) ~= 0

instance Approx M2 where
  x ~= y = sqrt (x || - || y) ~= 0
300 instance Approx M3 where
  x ~= y = sqrt (x || - || y) ~= 0

```

```

instance Approx M4 where
305   x ~ y = sqrt (x || - || y) ~ 0

{-
def gauss_jordan(m, eps = 1.0/(10**10)):
    """Puts given matrix (2D array) into the Reduced Row Echelon Form.
310     Returns True if successful, False if 'm' is singular.
    NOTE: make sure all the matrix items support fractions! Int matrix will NOT
        ↪ work!
    Written by Jarno Elonen in April 2005, released into Public Domain"""
    (h, w) = (len(m), len(m[0]))
    for y in range(0, h):
315         maxrow = y
         for y2 in range(y+1, h): # Find max pivot
             if abs(m[y2][y]) > abs(m[maxrow][y]):
                 maxrow = y2
         (m[y], m[maxrow]) = (m[maxrow], m[y])
320         if abs(m[y][y]) <= eps: # Singular?
             return False
         for y2 in range(y+1, h): # Eliminate column y
             c = m[y2][y] / m[y][y]
             for x in range(y, w):
325                 m[y2][x] -= m[y][x] * c
         for y in range(h-1, 0-1, -1): # Backsubstitute
             c = m[y][y]
             for y2 in range(0, y):
                 for x in range(w-1, y-1, -1):
330                     m[y2][x] -= m[y][x] * m[y2][y] / c
             m[y][y] /= c
             for x in range(h, w): # Normalize row y
                 m[y][x] /= c
         return True
335 -}

```

## 45 Vector.hs

```

{-# LANGUAGE MultiParamTypeClasses, TypeSynonymInstances #-}

module Vector where

5  --import Debug.Trace(trace)
import Graphics.UI.GLUT(GLfloat)

type R = GLfloat

10 data V1 = V1 !R deriving (Show, Eq, Ord)
data V2 = V2 !R !R deriving (Show, Eq, Ord)
data V3 = V3 !R !R !R deriving (Show, Eq, Ord)
data V4 = V4 !R !R !R !R deriving (Show, Eq, Ord)

15 class V a where
    o      :: a
    (^+)   :: a -> a -> a
    (^-)   :: a -> a -> a
    (*^)   :: R -> a -> a
20    (^*)   :: a -> R -> a

```

93

```

    (^^+^^) :: a -> a -> a
    (^^-^^) :: a -> a -> a
    (^^*^^) :: a -> a -> a
80    (*^^)   :: R -> a -> a
    (^^*)    :: a -> R -> a
    (^^/)    :: a -> R -> a
    mdot     :: a -> a -> R
    det      :: a -> R
85    inv     :: a -> a

(||-||) :: M a => a -> a -> R
u ||-|| v = let d = u ^^ - ^^ v in d `mdot` d

90    instance M M1 where
        i = M1 1
        M1 a11 ^^+^^ M1 b11 = M1 (a11 + b11)
        M1 a11 ^^ - ^^ M1 b11 = M1 (a11 - b11)
        M1 a11 ^^*^^ M1 b11 = M1 (a11 * b11)
95    a *^^ M1 b11 = M1 (a * b11)
        M1 a11 ^^* b = M1 (a11 * b)
        M1 a11 ^^/ b = M1 (a11 / b)
        M1 a11 `mdot` M1 b11 = (a11 * b11)
        det (M1 a11) = a11
100    inv (M1 a11) = M1 (1/a11)

instance M M2 where
    i = M2 1 0 0 1
    M2 a11 a12 a21 a22 ^^+^^ M2 b11 b12 b21 b22 =
105    M2 (a11 + b11) (a12 + b12) (a21 + b21) (a22 + b22)
    M2 a11 a12 a21 a22 ^^ - ^^ M2 b11 b12 b21 b22 =
        M2 (a11 - b11) (a12 - b12) (a21 - b21) (a22 - b22)
    M2 a11 a12 a21 a22 ^^*^^ M2 b11 b12 b21 b22 =
        M2 (a11*b11 + a12*b21) (a11*b12 + a12*b22) (a21*b11 + a22*b21) (a21*b12 +
        ↪ a22*b22)
110    a *^^ M2 b11 b12 b21 b22 = M2 (a * b11) (a * b12) (a * b21) (a * b22)
    M2 a11 a12 a21 a22 ^^* b = M2 (a11 * b) (a12 * b) (a21 * b) (a22 * b)
    M2 a11 a12 a21 a22 ^^/ b = M2 (a11 / b) (a12 / b) (a21 / b) (a22 / b)
    M2 a11 a12 a21 a22 `mdot` M2 b11 b12 b21 b22 =
        a11*b11 + a12*b12 + a21*b21 + a22*b22
115    det (M2 a11 a12 a21 a22) = a11 * a22 - a12 * a21
    inv a@(M2 a11 a12 a21 a22) = (M2 a22 (-a12) (-a21) a11) ^^/ det a

instance M M3 where
    i = M3 1 0 0 0 1 0 0 0 1
120    M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 ^^+^^ M3 b11 b12 b13 b21 b22 b23 b31 ↪
        ↪ b32 b33 =
        M3 (a11 + b11) (a12 + b12) (a13 + b13) (a21 + b21) (a22 + b22) (a23 + b23) (↪
        ↪ a31 + b31) (a32 + b32) (a33 + b33)
    M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 ^^ - ^^ M3 b11 b12 b13 b21 b22 b23 b31 ↪
        ↪ b32 b33 =
        M3 (a11 - b11) (a12 - b12) (a13 - b13) (a21 - b21) (a22 - b22) (a23 - b23) (↪
        ↪ a31 - b31) (a32 - b32) (a33 - b33)
    M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 ^^*^^ M3 b11 b12 b13 b21 b22 b23 b31 ↪
        ↪ b32 b33 =
125    M3 (a11*b11 + a12*b21 + a13*b31) (a11*b12 + a12*b22 + a13*b32) (a11*b13 + ↪
        ↪ a12*b23 + a13*b33)
        (a21*b11 + a22*b21 + a23*b31) (a21*b12 + a22*b22 + a23*b32) (a21*b13 + ↪

```

```

    ↪ a22*b23 + a23*b33)
    (a31*b11 + a32*b21 + a33*b31) (a31*b12 + a32*b22 + a33*b32) (a31*b13 + ↪
    ↪ a32*b23 + a33*b33)
a *^ M3 b11 b12 b13 b21 b22 b23 b31 b32 b33 =
  M3 (a * b11) (a * b12) (a * b13) (a * b21) (a * b22) (a * b23) (a * b31) (a ↪
  ↪ * b32) (a * b33)
130 M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 ^* b =
  M3 (a11 * b) (a12 * b) (a13 * b) (a21 * b) (a22 * b) (a23 * b) (a31 * b) (↪
  ↪ a32 * b) (a33 * b)
M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 ^/ b =
  M3 (a11 / b) (a12 / b) (a13 / b) (a21 / b) (a22 / b) (a23 / b) (a31 / b) (↪
  ↪ a32 / b) (a33 / b)
M3 a11 a12 a13 a21 a22 a23 a31 a32 a33 'mdot' M3 b11 b12 b13 b21 b22 b23 b31 ↪
  ↪ b32 b33 =
135 a11*b11 + a12*b12 + a13*b13 +
  a21*b21 + a22*b22 + a23*b23 +
  a31*b31 + a32*b32 + a33*b33
det (M3 a11 a12 a13 a21 a22 a23 a31 a32 a33) =
  let m11 = M2 a22 a23 a32 a33
140 m12 = M2 a21 a23 a31 a33
  m13 = M2 a21 a22 a31 a32
  in a11 * det m11 - a12 * det m12 + a13 * det m13
inv a@(M3 a11 a12 a13 a21 a22 a23 a31 a32 a33) =
  let m11 = a33 * a22 - a32 * a23
145 m12 = -(a33 * a12 - a32 * a13)
  m13 = a23 * a12 - a22 * a13
  m21 = -(a33 * a21 - a31 * a23)
  m22 = a33 * a11 - a31 * a13
  m23 = -(a23 * a11 - a21 * a13)
150 m31 = a32 * a21 - a31 * a22
  m32 = -(a32 * a11 - a31 * a12)
  m33 = a22 * a11 - a21 * a12
  in (M3 m11 m12 m13 m21 m22 m23 m31 m32 m33) ^/ det a

155 instance M M4 where
  i = M4 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1
  M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 ^+^ M4 ↪
  ↪ b11 b12 b13 b14 b21 b22 b23 b24 b31 b32 b33 b34 b41 b42 b43 b44 =
  M4 (a11 + b11) (a12 + b12) (a13 + b13) (a14 + b14)
    (a21 + b21) (a22 + b22) (a23 + b23) (a24 + b24)
160 (a31 + b31) (a32 + b32) (a33 + b33) (a34 + b34)
    (a41 + b41) (a42 + b42) (a43 + b43) (a44 + b44)
  M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 ^-^ M4 ↪
  ↪ b11 b12 b13 b14 b21 b22 b23 b24 b31 b32 b33 b34 b41 b42 b43 b44 =
  M4 (a11 - b11) (a12 - b12) (a13 - b13) (a14 - b14)
    (a21 - b21) (a22 - b22) (a23 - b23) (a24 - b24)
165 (a31 - b31) (a32 - b32) (a33 - b33) (a34 - b34)
    (a41 - b41) (a42 - b42) (a43 - b43) (a44 - b44)
  M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 ^*^ M4 ↪
  ↪ b11 b12 b13 b14 b21 b22 b23 b24 b31 b32 b33 b34 b41 b42 b43 b44 =
  M4 (a11*b11 + a12*b21 + a13*b31 + a14*b41) (a11*b12 + a12*b22 + a13*b32 + ↪
  ↪ a14*b42) (a11*b13 + a12*b23 + a13*b33 + a14*b43) (a11*b14 + a12*b24 + ↪
  ↪ a13*b34 + a14*b44)
    (a21*b11 + a22*b21 + a23*b31 + a24*b41) (a21*b12 + a22*b22 + a23*b32 + ↪
    ↪ a24*b42) (a21*b13 + a22*b23 + a23*b33 + a24*b43) (a21*b14 + a22*b24 ↪
    ↪ + a23*b34 + a24*b44)
170 (a31*b11 + a32*b21 + a33*b31 + a34*b41) (a31*b12 + a32*b22 + a33*b32 + ↪

```

```

    ↪ a34*b42) (a31*b13 + a32*b23 + a33*b33 + a34*b43) (a31*b14 + a32*b24 ↪
    ↪ + a33*b34 + a34*b44)
    (a41*b11 + a42*b21 + a43*b31 + a44*b41) (a41*b12 + a42*b22 + a43*b32 + ↪
    ↪ a44*b42) (a41*b13 + a42*b23 + a43*b33 + a44*b43) (a41*b14 + a42*b24 ↪
    ↪ + a43*b34 + a44*b44)
a * ^ M4 b11 b12 b13 b14 b21 b22 b23 b24 b31 b32 b33 b34 b41 b42 b43 b44 =
  M4 (a * b11) (a * b12) (a * b13) (a * b14)
    (a * b21) (a * b22) (a * b23) (a * b24)
175   (a * b31) (a * b32) (a * b33) (a * b34)
    (a * b41) (a * b42) (a * b43) (a * b44)
M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 ^ * b =
  M4 (a11 * b) (a12 * b) (a13 * b) (a14 * b)
    (a21 * b) (a22 * b) (a23 * b) (a24 * b)
180   (a31 * b) (a32 * b) (a33 * b) (a34 * b)
    (a41 * b) (a42 * b) (a43 * b) (a44 * b)
M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 ^ / b =
  M4 (a11 / b) (a12 / b) (a13 / b) (a14 / b)
    (a21 / b) (a22 / b) (a23 / b) (a24 / b)
185   (a31 / b) (a32 / b) (a33 / b) (a34 / b)
    (a41 / b) (a42 / b) (a43 / b) (a44 / b)
M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44 'mdot' M4 ↪
    ↪ b11 b12 b13 b14 b21 b22 b23 b24 b31 b32 b33 b34 b41 b42 b43 b44 =
    a11*b11 + a12*b12 + a13*b13 + a14*b14 +
    a21*b21 + a22*b22 + a23*b23 + a24*b24 +
190   a31*b31 + a32*b32 + a33*b33 + a34*b34 +
    a41*b41 + a42*b42 + a43*b43 + a44*b44
det (M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44) =
  let m11 = M3 a22 a23 a24 a32 a33 a34 a42 a43 a44
    m12 = M3 a21 a23 a24 a31 a33 a34 a41 a43 a44
195   m13 = M3 a21 a22 a24 a31 a32 a34 a41 a42 a44
    m14 = M3 a21 a22 a23 a31 a32 a33 a41 a42 a43
  in a11 * det m11 - a12 * det m12 + a13 * det m13 - a14 * det m14
inv (M4 a11 a12 a13 a14 a21 a22 a23 a24 a31 a32 a33 a34 a41 a42 a43 a44) =
  let a = M2 a11 a12 a21 a22
200   b = M2 a13 a14 a23 a24
    c = M2 a31 a32 a41 a42
    d = M2 a33 a34 a43 a44
    a1 = inv a
    ca1 = c ^ ^ * ^ ^ a1
205   dcab1 = inv (d ^ ^ - ^ ^ (ca1 ^ ^ * ^ ^ b))
    albdcab1 = a1 ^ ^ * ^ ^ (b ^ ^ * ^ ^ dcab1)
    M2 m11 m12 m21 m22 = a1 ^ ^ + ^ ^ (albdcab1 ^ ^ * ^ ^ ca1)
    M2 m13 m14 m23 m24 = (-1) * ^ ^ albdcab1
    M2 m31 m32 m41 m42 = (-1) * ^ ^ (dcab1 ^ ^ * ^ ^ ca1)
210   M2 m33 m34 m43 m44 = dcab1
  in M4 m11 m12 m13 m14 m21 m22 m23 m24 m31 m32 m33 m34 m41 m42 m43 m44

class MV m v where
  ( ^ ^ * ^ ) :: m -> v -> v
215   ( ^ ^ ! ) :: v -> v -> m

reflector :: (M m, V v, MV m v) => v -> m
reflector v = i ^ ^ - ^ ^ (2 * ^ ^ (v ^ ^ * ^ ! v))

220 instance MV M1 V1 where
  M1 m ^ ^ * ^ V1 v = V1 (m * v)
  V1 a ^ ^ * ^ ! V1 b = M1 (a * b)

```

```

instance MV M2 V2 where
225   M2 m11 m12 m21 m22 ^^*^ V2 v1 v2 =
       V2 (m11*v1 + m12*v2)
          (m21*v1 + m22*v2)
       V2 a1 a2 ^*^! V2 b1 b2 =
       M2 (a1 * b1) (a1 * b2)
230         (a2 * b1) (a2 * b2)

instance MV M3 V3 where
       M3 m11 m12 m13 m21 m22 m23 m31 m32 m33 ^^*^ V3 v1 v2 v3 =
       V3 (m11*v1 + m12*v2 + m13*v3)
235         (m21*v1 + m22*v2 + m23*v3)
          (m31*v1 + m32*v2 + m33*v3)
       V3 a1 a2 a3 ^*^! V3 b1 b2 b3 =
       M3 (a1 * b1) (a1 * b2) (a1 * b3)
          (a2 * b1) (a2 * b2) (a2 * b3)
240         (a3 * b1) (a3 * b2) (a3 * b3)

instance MV M4 V4 where
       M4 m11 m12 m13 m14 m21 m22 m23 m24 m31 m32 m33 m34 m41 m42 m43 m44 ^^*^ V4 v1 v2 v3 v4 =
       V4 (m11*v1 + m12*v2 + m13*v3 + m14*v4)
245         (m21*v1 + m22*v2 + m23*v3 + m24*v4)
          (m31*v1 + m32*v2 + m33*v3 + m34*v4)
          (m41*v1 + m42*v2 + m43*v3 + m44*v4)
       V4 a1 a2 a3 a4 ^*^! V4 b1 b2 b3 b4 =
       M4 (a1 * b1) (a1 * b2) (a1 * b3) (a1 * b4)
250         (a2 * b1) (a2 * b2) (a2 * b3) (a2 * b4)
          (a3 * b1) (a3 * b2) (a3 * b3) (a3 * b4)
          (a4 * b1) (a4 * b2) (a4 * b3) (a4 * b4)

cross4 :: V4 -> V4 -> V4 -> V4
255 cross4 (V4 u0 u1 u2 u3) (V4 v0 v1 v2 v3) (V4 w0 w1 w2 w3) =
{-
   let vw01 = v0 * w1 - v1 * w0
       vw02 = v0 * w2 - v2 * w0
       vw03 = v0 * w3 - v3 * w0
260       vw12 = v1 * w2 - v2 * w1
       vw13 = v1 * w3 - v3 * w1
       vw23 = v2 * w3 - v3 * w2
       r0 = u1 * vw23 - u2 * vw13 + u3 * vw12
       r1 = - u0 * vw23 + u2 * vw03 - u3 * vw02
265       r2 = u0 * vw13 - u1 * vw03 + u3 * vw02
       r3 = - u0 * vw12 + u1 * vw02 - u2 * vw01
   in V4 r0 r1 r2 r3
-}
270 let m0 = M3 u1 u2 u3 v1 v2 v3 w1 w2 w3
       m1 = M3 u0 u2 u3 v0 v2 v3 w0 w2 w3
       m2 = M3 u0 u1 u3 v0 v1 v3 w0 w1 w3
       m3 = M3 u0 u1 u2 v0 v1 v2 w0 w1 w2
   in V4 (det m0) (-(det m1)) (det m2) (-(det m3))

275
epsilon :: R
epsilon = 1e-3

```

```

280 class Approx a where
    (~=) :: a -> a -> Bool

instance Approx R where
    x ~= y = abs (x - y) < epsilon
285
instance Approx V1 where
    x ~= y = sqrt (x |-| y) ~= 0

instance Approx V2 where
290     x ~= y = sqrt (x |-| y) ~= 0

instance Approx V3 where
    x ~= y = sqrt (x |-| y) ~= 0

295 instance Approx V4 where
    x ~= y = sqrt (x |-| y) ~= 0

instance Approx M1 where
    x ~= y = sqrt (x || -|| y) ~= 0
300
instance Approx M2 where
    x ~= y = sqrt (x || -|| y) ~= 0

instance Approx M3 where
305     x ~= y = sqrt (x || -|| y) ~= 0

instance Approx M4 where
    x ~= y = sqrt (x || -|| y) ~= 0

310 {-
def gauss_jordan(m, eps = 1.0/(10**10)):
    """Puts given matrix (2D array) into the Reduced Row Echelon Form.
    Returns True if successful, False if 'm' is singular.
    NOTE: make sure all the matrix items support fractions! Int matrix will NOT
        ↪ work!
315     Written by Jarno Elonen in April 2005, released into Public Domain"""
    (h, w) = (len(m), len(m[0]))
    for y in range(0,h):
        maxrow = y
        for y2 in range(y+1, h): # Find max pivot
320             if abs(m[y2][y]) > abs(m[maxrow][y]):
                maxrow = y2
        (m[y], m[maxrow]) = (m[maxrow], m[y])
        if abs(m[y][y]) <= eps: # Singular?
            return False
325     for y2 in range(y+1, h): # Eliminate column y
        c = m[y2][y] / m[y][y]
        for x in range(y, w):
            m[y2][x] -= m[y][x] * c
    for y in range(h-1, 0-1, -1): # Backsubstitute
330     c = m[y][y]
        for y2 in range(0,y):
            for x in range(w-1, y-1, -1):
                m[y2][x] -= m[y][x] * m[y2][y] / c
            m[y2][y] /= c

```

```
335     for x in range(h, w):      # Normalize row y
        m[y][x] /= c
    return True
-}
```