

sp-at-adf

Claude Heiland-Allen

2013–2018

Contents

1	chapter-2/build_cuboloid_mesh.hs	4
2	chapter-2/colours.c	6
3	chapter-2/.gitignore	7
4	chapter-2/Makefile	8
5	chapter-2/notes.txt	8
6	chapter-2/oloid.cc	10
7	chapter-2/oloid.gnuplot	24
8	LICENSE	25
9	prototype/anaglyph-help.pd	25
10	prototype/anaglyph.pd	42
11	prototype/ball-sound.pd	45
12	prototype/ball-sounds.pd	47
13	prototype/encode.sh	48
14	prototype/geo-pipe.pd	48
15	prototype/ground.pd	50
16	prototype/lightning-0.jpg	54
17	prototype/lightning-1.jpg	55
18	prototype/lightning-2.jpg	55
19	prototype/lightning-3.jpg	56
20	prototype/lightning-4.jpg	56
21	prototype/lightning-5.jpg	57
22	prototype/lightning-6.jpg	57
23	prototype/lightning-7.jpg	58
24	prototype/lightning-8.jpg	58
25	prototype/lightning-9.jpg	59
26	prototype/lightning.pd	59
27	prototype/mountains.pd	60
28	prototype/mountainss.pd	62
29	prototype/multi-texture-1.pd	63
30	prototype/multi-texture.pd	63
31	prototype/once.pd	64
32	prototype/repeat-gemlist.pd	64
33	prototype/snake.pd	65
34	prototype/snake-shape.pd	66
35	prototype/snakes.pd_lua	68
36	prototype/starfield.pd	70
37	prototype/start.sh	71
38	prototype/teapot.obj	71
39	prototype/vec3-add.pd	122
40	prototype/vec3-length.pd	122
41	prototype/vec3-normalize.pd	123
42	prototype/vec3-scale.pd	123

43	prototype/with-blend.pd	124
44	prototype/with-lighting.pd	124
45	prototype/without-lighting.pd	124
46	prototype/worm-sound.pd	124
47	prototype/worm-sounds.pd	126
48	sketches/byzantine/byzantine.cc	127
49	sketches/byzantine/.gitignore	130
50	sketches/byzantine/Makefile	130
51	sketches/colour/colour.c	130
52	sketches/colour/Makefile	130
53	sketches/craterscape/craterscape.cpp	131
54	sketches/craterscape/.gitignore	143
55	sketches/craterscape/Makefile	143
56	sketches/craters/craters.c	143
57	sketches/craters/.gitignore	147
58	sketches/craters/Makefile	147
59	sketches/dolly/3D.frag	147
60	sketches/dolly/BufferShader.frag	151
61	sketches/dolly/DE-Raytracer.frag	152
62	sketches/dolly/dolly.frag	159
63	sketches/juggling/juggle-help.pd	159
64	sketches/juggling/juggle.pd.lua	161
65	sketches/juggling/juggler-help.pd	163
66	sketches/juggling/juggler.pd.lua	163
67	sketches/lightning/.gitignore	165
68	sketches/lightning/lightning.c	165
69	sketches/lightning/lightning.sh	168
70	sketches/lightning/Makefile	168
71	sketches/mountains/.gitignore	168
72	sketches/mountains/Makefile	168
73	sketches/mountains/mountains.c	168
74	sketches/mountains/mountains.sh	169
75	sketches/oloid/.gitignore	169
76	sketches/oloid/Makefile	169
77	sketches/oloid/oloid.hs	169
78	sketches/oloid/oloid.pd	171
79	sketches/prism/.gitignore	174
80	sketches/prism/Makefile	174
81	sketches/prism/prism.frag	174
82	sketches/prism/prism-help.pd	175
83	sketches/prism/prism.hs	175
84	sketches/prism/prism.pd	177
85	sketches/prism/prism.vert	178
86	sketches/thunder/distance.pd	178
87	sketches/thunder/strike-pattern.pd	179
88	sketches/thunder/strike-sound.pd	180
89	sketches/thunder/thunder4.pd	181
90	sketches/thunder/udly.pd	185

1 chapter-2/build_cuboloid_mesh.hs

```

import Control.Monad (forM_)
import Data.Ix (inRange)
import Data.List (intercalate)
import Data.Map.Strict (Map)
5 import qualified Data.Map.Strict as Map
import Data.Word (Word32)
import Foreign (with)
import System.IO (withBinaryFile, IOMode(WriteMode), hPutBuf)

10 data Vertex = V{ v't, v'v, v'ox, v'oy, v'oz, v'cx, v'cy, v'cz, v'nx, v'ny, v'nz }
    ↪ :: !Double, v'e :: !Bool }
    deriving (Read, Show, Eq, Ord)

vertex :: Double -> Double -> Vertex
vertex t v = V t v ox oy oz cx cy cz nx ny nz False
15 where
    -- oloid param
    u = 2 * pi / 3 * signum t * (3 * abs t - 2 * abs t ** 1.5)
    c = (-1/2) 'max' cos u
    s = signum t * sqrt (0 'max' (1 - c * c))
20 -- oloid arcs
    ox0 = s
    oy0 = - c - 1/2
    oz0 = 0
    ox1 = 0
25 oy1 = 1 / (1 + c) - 1/2
    oz1 = signum v * sqrt (1 + 2 * c) / (1 + c)
    -- oloid surface
    ox = ox0 * (1 - abs v) + abs v * ox1
    oy = oy0 * (1 - abs v) + abs v * oy1
30 oz = oz0 * (1 - abs v) + abs v * oz1
    -- sphere surface
    sl = sqrt (ox * ox + oy * oy + oz * oz)
    sx = ox / sl
    sy = oy / sl
35 sz = oz / sl
    -- cube surface
    cl = abs sx 'max' abs sy 'max' abs sz
    cx = sx / cl
    cy = sy / cl
40 cz = sz / cl
    -- cube normal
    nl = abs cx 'max' abs cy 'max' abs cz
    nx = if abs cx == nl then signum cx else 0
    ny = if abs cy == nl then signum cy else 0
45 nz = if abs cz == nl then signum cz else 0

type VID = (Int, Int)

data Mesh = M{ m'n :: !Int, m'v :: !(Map VID Vertex) }
50 deriving (Read, Show, Eq, Ord)

mesh :: Int -> Mesh
mesh n = M n (Map.fromList [ ((i, j), vertex (f i) (f j)) | i <- [0 .. n], j <- ↪
    ↪ [0 .. n] ])

```

```

where
55   f t = (fromIntegral t / fromIntegral n) * 2 - 1

edges :: Int -> [(VID, VID)]
edges n =
  [ ((i, j), v)
60   | i <- [0 .. n]
    , j <- [0 .. n]
    , v <- [ (i, j + 1), (i + 1, j), (i + 1, j + 1) ]
    , inRange ((0, 0), (n, n)) v
  ]
65

crossesCubeEdge :: Vertex -> Vertex -> Bool
crossesCubeEdge V{ v'nx = ax, v'ny = ay, v'nz = az } V{ v'nx = bx, v'ny = by, v'nz = bz } =
  ax * bx + ay * by + az * bz == 0

70 splitEdge :: Vertex -> Vertex -> Vertex
splitEdge x y
  | v'e x || v'e y = mid x y
  | crossesCubeEdge x y = (search 64 x (mid x y) y) { v'e = True }
  | otherwise = mid x y
75 where
  mid a b = vertex ((v't a + v't b) / 2) ((v'v a + v'v b) / 2)
  search 0 _ b _ = b
  search n a b c = case (crossesCubeEdge a b, crossesCubeEdge b c) of
    (False, False) -> b
80    (True, False) -> search (n - 1) a (mid a b) b
    (False, True) -> search (n - 1) b (mid b c) c
    (True, True) -> search (n - 1) a (mid a b) b -- crosses 2 edges, near ↗
    ↪ corner?

refine :: Mesh -> Mesh
85 refine (M n0 vs0) = M n vs
  where
    n = 2 * n0
    vs1 = Map.mapKeysMonotonic (\(i,j) -> (2 * i, 2 * j)) vs0
    vs2 = Map.fromList
90    [ ((i0 + i1, j0 + j1), splitEdge (vs0 Map.! ij0) (vs0 Map.! ij1))
      | (ij0@(i0, j0), ij1@(i1, j1)) <- edges n0
    ]
    vs = Map.union vs1 vs2

95 canonical :: Int -> VID -> VID
canonical n (i, j)
  | i == 0 = (i, min j (n - j))
  | j == 0 = (min i (n - i), j)
  | i == n = (i, min j (n - j))
100 | j == n = (min i (n - i), j)
  | otherwise = (i, j)

index :: Int -> VID -> Int
index n (i, j) = i + (n + 1) * j
105

triangles :: Int -> [Int]
triangles n = map (index n . canonical n) . concat $
  [ [ (i, j), (i + 1, j), (i + 1, j + 1) ]

```

```

110     , (i, j), (i + 1, j + 1), (i, j + 1) ]
    | i <- [0 .. n - 1]
    , j <- [0 .. n - 1]
    ]

vertices :: Map VID Vertex -> [Double]
115 vertices vs = concat
    [ [ ox, oy, oz, cx, cy, cz ] | V - - ox oy oz cx cy cz - - - - <- Map.elems vs
      ↵ ]

pretty :: Mesh -> String
pretty (M n vs) =
120     "GLuint cuboloid_indices [] = {\n " ++
        intercalate "," (map show (triangles n)) ++
        "\n};\nGLfloat cuboloid_vertices [] = {\n " ++
        intercalate "," (map show (vertices vs)) ++
        "\n};\n"
125
main :: IO ()
main = do
    let n = 1024
    withBinaryFile "cuboloid.u32" WriteMode $ \h -> do
130         forM_ (map (fromIntegral :: Int -> Word32) . triangles . (2 *) $ n) $ \w -> ↵
            ↵ do
                with w $ \p -> hPutBuf h p 4
    withBinaryFile "cuboloid.f32" WriteMode $ \h -> do
        forM_ (map (realToFrac :: Double -> Float) . vertices . m'v . refine . mesh ↵
            ↵ $ n) $ \w -> do
                with w $ \p -> hPutBuf h p 4
135
{-
-- for gnuplotting
main
    = writeFile "mesh-rgb.dat"
140    . unlines
    . map (\v -> unwords [show $ v't v, show $ v'v v, show $ rgb (v'nx v) (v'ny v) ↵
        ↵ (v'nz v)])
    . filter v'e
    . Map.elems
    . m'v
145    . refine
    . mesh
    $ 128
    where
        rgb r g b = 65536 * c r + 256 * c g + c b
150        c x = round $ 0 'max' 255 * (x + 1) / 2 'min' 255
-}

```

2 chapter-2/colours.c

```

#include <math.h>
#include <stdio.h>
#include <stdlib.h>

5 #define twopi 6.283185307179586
#define maxcount 2000

```

```

struct { float h, s, v, r, g, b; } c[maxcount];

10 int main(int argc, char **argv) {
    int count = 0;
    for (int i = 0; i < maxcount; ++i) {
        count = i;
        if (6 != scanf("%f %f %f %f %f %f\n", &c[i].h, &c[i].v, &c[i].s, &c[i].r, &c[i].g, &c[i].b)) { break; };
15     c[i].h /= 100.0f;
    }
    int fs = 499 * 60;
    for (int f = 0; f < fs; ++f) {
        float x = f / (float) fs;
20     float h = 10 * fmod(8 * x, 1);
        float s = 15 - 5 * cos(twopi * 7 * x);
        float v = 8;
        float r = 0;
        float g = 0;
25     float b = 0;
        float a = 0;
        for (int i = 0; i < count; ++i) {
            float dh = fminf(fabsf(c[i].h - h), 10.0f - fabsf(c[i].h - h));
            float ds = c[i].s - s;
30     float dv = c[i].v - v;
            float d = dh * dh + ds * ds + dv * dv;
            float w = exp(-d / 4.0);
            r += c[i].r * w;
            g += c[i].g * w;
35     b += c[i].b * w;
            a += w;
        }
        r /= a;
        g /= a;
40     b /= a;
        printf("%f %f %f\n", r, g, b);
    }
    return 0;
}

```

3 chapter-2/.gitignore

```

*.mkv
*.mp4
*.wav
*.m4a
5 *.f32
*.u32
*.hi
*.o
*.sh
10 *.zip
*.png
*/
cuboloid
build_cuboloid_mesh
15 playlist.txt

```

4 chapter-2/Makefile

```
cuboloid: oloid.cc cuboloid.f32
      g++ -std=c++11 -pedantic -Wall -Wextra -O3 -march=native -o cuboloid \
      ↪ oloid.cc -lGL -lGLEW -lglfw
```

```

cuboloid.f32: build_cuboloid_mesh
5      ./build_cuboloid_mesh

```

```
build_cuboloid_mesh: build_cuboloid_mesh.hs
                    ghc -O2 build_cuboloid_mesh.hs
```

5 chapter-2/notes.txt

```
<http://www.rit.edu/cos/colorscience/rc.munsell-renotation.php>  
<http://www.rit-mcsl.org/MunsellRenotation/real_sRGB.xls>  
cat real_sRGB.csv |  
sed 's|.*,\(.*\) , \(.*\) , \(.*\) , .* , .* , .* , .* , .* , .* , .* , .* , .* , \(.*\) , \(.*\) , \(.*\) \2  
↳ , .*, *, * ,*\1 \2 \3 \4 \5 \6|' |  
5 sed -e 's|YR| 2|' -e 's|GY| 4|' -e 's|BG| 6|' -e 's|PB| 8|' -e 's|RP| 10|' -e 's|  
↳ |R| 1|' -e 's|Y| 3|' -e 's|G| 5|' -e 's|B| 7|' -e 's|P| 9|' |  
ghc -e 'interact (unlines . map unwords . map (map show) . (\xs -> [ [(b-1) * ↵  
↳ 100.0 + a, c, d, e, f, g] | [a,b,c,d,e,f,g] <- xs ]) . map (map read) . ↵  
↳ map words . lines)' > hvs2srgb.txt  
  
./cuboloid | ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x360 -r 25 -i - -vb 2M ↵  
↳ output.ogv  
  
10 mkfifo test.yuv  
./cuboloid > test.yuv &  
x264 --demuxer lavf --input-csp yuv420p --input-range tv --colorprim bt709 --↵  
↳ transfer bt709 --colormatrix bt709 --sar 1:1 --fps 25 --profile high --↵  
↳ bluray-compat --crf 12 --input-res 640x360 -o output.mkv test.yuv  
  
./cuboloid | ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x360 -r 25 -i - -pass 1 ↵  
↳ -c:v libx264 -b:v 760k -bt 760k -y output1.mkv  
15 ./cuboloid | ffmpeg -f rawvideo -pix_fmt yuv420p -s:v 640x360 -r 25 -i - -pass 2 ↵  
↳ -c:v libx264 -b:v 760k -bt 760k -y output.mkv  
  
[libx264 @ 0x15b0060] frame MB size (120x75) > level limit (8192)  
[libx264 @ 0x15b0060] DPB size (4 frames, 36000 mbs) > level limit (3 frames, ↵  
↳ 32768 mbs)  
20 [libx264 @ 0x15b0060] MB rate (540000) > level limit (245760)  
[libx264 @ 0x15b0060] using cpu capabilities: MMX2 SSE2Fast LZCNT  
[libx264 @ 0x15b0060] profile High, level 4.0  
[libx264 @ 0x15b0060] 264 - core 142 r243l a5831aa - H.264/MPEG-4 AVC codec - ↵  
↳ Copyleft 2003-2014 - http://www.videolan.org/x264.html - options: cabac=1 ↵  
↳ ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1 psy_rd↵  
↳ =1.00:0.00 mixed_ref=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 qcm=0 ↵  
↳ deadzone=21,11 fast_pskip=1 chroma_qp_offset=-2 threads=6 ↵  
↳ lookahead_threads=1 sliced_threads=0 nr=0 decimate=1 interlaced=0 ↵  
↳ bluray_compat=0 constrained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b_bias↵  
↳ =0 direct=1 weightb=1 open_gop=0 weightp=2 keyint=250 keyint_min=25 ↵  
↳ scenecut=40 intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=18.0 qcomp↵  
↳ =0.60 qpmin=0 qpmax=69 qpstep=4 ip_ratio=1.40 aq=1:1.00
```



```

25  [libx264 @ 0x21f6060] frame MB size (120x75) > level limit (8192)
    [libx264 @ 0x21f6060] DPB size (4 frames, 36000 mbs) > level limit (3 frames, ↵
        ↵ 32768 mbs)
    [libx264 @ 0x21f6060] MB rate (540000) > level limit (245760)
    [libx264 @ 0x21f6060] using cpu capabilities: MMX2 SSE2Fast LZCNT
    [libx264 @ 0x21f6060] profile High, level 4.1
30  [libx264 @ 0x21f6060] 264 - core 142 r2431 a5831aa - H.264/MPEG-4 AVC codec - ↵
        ↵ Copyleft 2003-2014 - http://www.videolan.org/x264.html - options: cabac=1 ↵
        ↵ ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1 psy_rd↵
        ↵ =1.00:0.00 mixed_ref=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0 ↵
        ↵ deadzone=21,11 fast_pskip=1 chroma_qp_offset=-2 threads=6 ↵
        ↵ lookahead_threads=1 sliced_threads=0 nr=0 decimate=1 interlaced=0 ↵
        ↵ bluray_compat=0 constrained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b_bias↵
        ↵ =0 direct=1 weightb=1 open_gop=0 weightp=2 keyint=250 keyint_min=25 ↵
        ↵ scenecut=40 intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=18.0 qcomp↵
        ↵ =0.60 qpmin=0 qpmax=69 qpstep=4 ip_ratio=1.40 aq=1:1.00

    [libx264 @ 0x116b060] frame MB size (120x75) > level limit (8704)
    [libx264 @ 0x116b060] DPB size (4 frames, 36000 mbs) > level limit (3 frames, ↵
        ↵ 34816 mbs)
    [libx264 @ 0x116b060] MB rate (540000) > level limit (522240)
35  [libx264 @ 0x116b060] using cpu capabilities: MMX2 SSE2Fast LZCNT
    [libx264 @ 0x116b060] profile High, level 4.2
    [libx264 @ 0x116b060] 264 - core 142 r2431 a5831aa - H.264/MPEG-4 AVC codec - ↵
        ↵ Copyleft 2003-2014 - http://www.videolan.org/x264.html - options: cabac=1 ↵
        ↵ ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1 psy_rd↵
        ↵ =1.00:0.00 mixed_ref=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0 ↵
        ↵ deadzone=21,11 fast_pskip=1 chroma_qp_offset=-2 threads=6 ↵
        ↵ lookahead_threads=1 sliced_threads=0 nr=0 decimate=1 interlaced=0 ↵
        ↵ bluray_compat=0 constrained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b_bias↵
        ↵ =0 direct=1 weightb=1 open_gop=0 weightp=2 keyint=250 keyint_min=25 ↵
        ↵ scenecut=40 intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=18.0 qcomp↵
        ↵ =0.60 qpmin=0 qpmax=69 qpstep=4 ip_ratio=1.40 aq=1:1.00

    [libx264 @ 0x1162060] frame MB size (120x75) > level limit (8192)
40  [libx264 @ 0x1162060] DPB size (4 frames, 36000 mbs) > level limit (3 frames, ↵
        ↵ 32768 mbs)
    [libx264 @ 0x1162060] MB rate (540000) > level limit (245760)
    [libx264 @ 0x1162060] using cpu capabilities: MMX2 SSE2Fast LZCNT
    [libx264 @ 0x1162060] profile High 4:4:4 Predictive, level 4.0, 4:4:4 8-bit
    [libx264 @ 0x1162060] 264 - core 142 r2431 a5831aa - H.264/MPEG-4 AVC codec - ↵
        ↵ Copyleft 2003-2014 - http://www.videolan.org/x264.html - options: cabac=1 ↵
        ↵ ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1 psy_rd↵
        ↵ =1.00:0.00 mixed_ref=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0 ↵
        ↵ deadzone=21,11 fast_pskip=1 chroma_qp_offset=4 threads=6 lookahead_threads↵
        ↵ =1 sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_compat=0 ↵
        ↵ constrained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b_bias=0 direct=1 ↵
        ↵ weightb=1 open_gop=0 weightp=2 keyint=250 keyint_min=25 scenecut=40 ↵
        ↵ intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=18.0 qcomp=0.60 qpmin↵
        ↵ =0 qpmax=69 qpstep=4 ip_ratio=1.40 aq=1:1.00

45  [libx264 @ 0xc41060] frame MB size (120x75) > level limit (8192)
    [libx264 @ 0xc41060] DPB size (4 frames, 36000 mbs) > level limit (3 frames, ↵
        ↵ 32768 mbs)
    [libx264 @ 0xc41060] MB rate (540000) > level limit (245760)
    [libx264 @ 0xc41060] using cpu capabilities: MMX2 SSE2Fast LZCNT

```

```

50  [libx264 @ 0xc41060] profile High 4:4:4 Predictive , level 4.1, 4:4:4 8-bit
    [libx264 @ 0xc41060] 264 - core 142 r2431 a5831aa - H.264/MPEG-4 AVC codec - ↵
        ↳ Copley 2003-2014 - http://www.videolan.org/x264.html - options: cabac=1 ↵
        ↳ ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1 psy_rd↵
        ↳ =1.00:0.00 mixed_ref=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0 ↵
        ↳ deadzone=21,11 fast_pskip=1 chroma_qp_offset=4 threads=6 lookahead_threads↵
        ↳ =1 sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_compat=0 ↵
        ↳ constrained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b_bias=0 direct=1 ↵
        ↳ weightb=1 open_gop=0 weightp=2 keyint=250 keyint_min=25 scenecut=40 ↵
        ↳ intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=18.0 qcomp=0.60 qpmin↵
        ↳ =0 qpmax=69 qpstep=4 ip_ratio=1.40 aq=1:1.00

    [libx264 @ 0x14cd060] frame MB size (120x75) > level limit (8704)
    [libx264 @ 0x14cd060] DPB size (4 frames, 36000 mbs) > level limit (3 frames, ↵
        ↳ 34816 mbs)
55  [libx264 @ 0x14cd060] MB rate (540000) > level limit (522240)
    [libx264 @ 0x14cd060] using cpu capabilities: MMX2 SSE2Fast LZCNT
    [libx264 @ 0x14cd060] profile High 4:4:4 Predictive , level 4.2, 4:4:4 8-bit
    [libx264 @ 0x14cd060] 264 - core 142 r2431 a5831aa - H.264/MPEG-4 AVC codec - ↵
        ↳ Copley 2003-2014 - http://www.videolan.org/x264.html - options: cabac=1 ↵
        ↳ ref=3 deblock=1:0:0 analyse=0x3:0x113 me=hex subme=7 psy=1 psy_rd↵
        ↳ =1.00:0.00 mixed_ref=1 me_range=16 chroma_me=1 trellis=1 8x8dct=1 cqm=0 ↵
        ↳ deadzone=21,11 fast_pskip=1 chroma_qp_offset=4 threads=6 lookahead_threads↵
        ↳ =1 sliced_threads=0 nr=0 decimate=1 interlaced=0 bluray_compat=0 ↵
        ↳ constrained_intra=0 bframes=3 b_pyramid=2 b_adapt=1 b_bias=0 direct=1 ↵
        ↳ weightb=1 open_gop=0 weightp=2 keyint=250 keyint_min=25 scenecut=40 ↵
        ↳ intra_refresh=0 rc_lookahead=40 rc=crf mbtree=1 crf=18.0 qcomp=0.60 qpmin↵
        ↳ =0 qpmax=69 qpstep=4 ip_ratio=1.40 aq=1:1.00

60  ffmpeg -i 2400f.wav -b:a 256k -strict experimental 1920x1200p60x1e.aac
    ffmpeg -i 2395f.wav -b:a 256k -strict experimental 1920x1200p60x25.aac

```

6 chapter-2/oloid.cc

```

#include <GL/glew.h>
#include <GLFW/glfw3.h>
#include <glm/glm.hpp>
#include <glm/gtc/matrix_transform.hpp>
5  #include <stdio.h>
    #include <stdlib.h>

    // linear interpolation by Robert Munafo mrob.com
    static float lin_interp
10  ( float x
    , float domain_low
    , float domain_hi
    , float range_low
    , float range_hi
15  ) {
    if ((x >= domain_low) && (x <= domain_hi)) {
        x = (x - domain_low) / (domain_hi - domain_low);
        x = range_low + x * (range_hi - range_low);
    }
20  return x;
}

```

```

// perceptually balanced hue adjustment by Robert Munafo mrob.com
static float pvp_adjust_3(float x) {
25   x = lin_interp(x, 0.00, 0.125, -0.050, 0.090);
    x = lin_interp(x, 0.125, 0.25, 0.090, 0.167);
    x = lin_interp(x, 0.25, 0.375, 0.167, 0.253);
    x = lin_interp(x, 0.375, 0.50, 0.253, 0.383);
    x = lin_interp(x, 0.50, 0.625, 0.383, 0.500);
30   x = lin_interp(x, 0.625, 0.75, 0.500, 0.667);
    x = lin_interp(x, 0.75, 0.875, 0.667, 0.800);
    x = lin_interp(x, 0.875, 1.00, 0.800, 0.950);
    return(x);
}
35
static void hsv2rgb
( float h
, float s
, float v
40 , float *rp
, float *gp
, float *bp
) {
    float i, f, p, q, t, r, g, b;
45   if (s == 0.0) {
        r = v;
        g = v;
        b = v;
    } else {
50     h = pvp_adjust_3(h);
        h = h - floor(h);
        h = h * 6.0;
        i = floor(h);
        f = h - i;
55     p = v*(1.0 - s);
        q = v*(1.0 - (s*f));
        t = v*(1.0 - (s*(1.0 - f)));
        if (i < 1.0) { r = v; g = t; b = p; } else
        if (i < 2.0) { r = q; g = v; b = p; } else
60     if (i < 3.0) { r = p; g = v; b = t; } else
        if (i < 4.0) { r = p; g = q; b = v; } else
        if (i < 5.0) { r = t; g = p; b = v; } else
        { r = v; g = p; b = q; }
    }
65   *rp = r;
    *gp = g;
    *bp = b;
}

70   static const double pi = 3.141592653589793;
    static const double twopi = 6.283185307179586;

    static const char *morph_vertex = R"(
#version 410 core
75   layout(location = 0) in vec3 oloid;
    layout(location = 1) in vec3 cube;
    uniform float morph;
    void main(void) {
        gl_Position = vec4(mix(oloid, cube / pow(8.0/3.052418468, 1.0/3.0), morph), 1)

```

```

    ↪ 1.0);
80 }
    )”);

    static const char *morph_geometry = R”(
#version 410 core
85 layout(triangles) in;
    layout(triangle_strip, max_vertices = 3) out;
    out vec3 surface;
    out vec3 normal;
    uniform mat4 modelMatrix;
90 uniform mat4 projectionMatrix;
    uniform mat3 normalMatrix;
    void main(void) {
        vec3 d1 = gl_in[1].gl_Position.xyz - gl_in[0].gl_Position.xyz;
        vec3 d2 = gl_in[2].gl_Position.xyz - gl_in[0].gl_Position.xyz;
95 vec3 n = normalMatrix * -normalize(cross(normalize(d1), normalize(d2)));
        vec4 p0 = modelMatrix * gl_in[0].gl_Position;
        vec4 p1 = modelMatrix * gl_in[1].gl_Position;
        vec4 p2 = modelMatrix * gl_in[2].gl_Position;
        p0.w *= 3.0 / -p0.z;
100 p1.w *= 3.0 / -p1.z;
        p2.w *= 3.0 / -p2.z;
        vec3 s0 = p0.xyz / p0.w;
        vec3 s1 = p1.xyz / p1.w;
        vec3 s2 = p2.xyz / p2.w;
105 p0 = projectionMatrix * p0;
        p1 = projectionMatrix * p1;
        p2 = projectionMatrix * p2;
        surface = s0;
        normal = n;
110 gl_Position = p0;
        EmitVertex();
        surface = s1;
        normal = n;
        gl_Position = p1;
115 EmitVertex();
        surface = s2;
        normal = n;
        gl_Position = p2;
        EmitVertex();
120 EndPrimitive();
    }
    )”);

// <https://en.wikipedia.org/wiki/Specular\_highlight#Cook.E2.80.93Torrance\_model>
    ↪ >
125 // <http://inst.eecs.berkeley.edu/~cs283/sp13/lectures/cookpaper.pdf>
    static const char *morph_fragment = R”(
#version 410 core
    layout(location = 0, index = 0) out vec4 colour;
    const float pi = 3.141592653589793;
130 uniform vec3 eye;
    uniform vec3 material;
    uniform float ambient;
    uniform float diffuse;
    uniform float specular;

```

```

135  uniform float index;
    uniform float roughness;
    uniform float alpha;
    uniform float scale;
    in vec3 surface;
140  in vec3 normal;
    void main(void) {
        vec3 V = normalize(eye - surface);
        V.xy *= -1.0;
        vec3 N = normalize(normal);
145  float diff = ambient;
        float spec = 0.0;
        float q = 1.0;
        float vn = dot(V, N);
        if (vn < 0.0) {
150      discard;
        } else {
            for (int i = -1; i <= 1; i += 1) {
                for (int j = -1; j <= 1; j += 1) {
                    vec3 light = scale * 32.0 * vec3(float(i), float(j), 0.0);
155      vec3 L = normalize(light - surface);
                    vec3 H = normalize(L + V);
                    float k = 0.0;
                    float ln = max(0.0, dot(L, N));
                    if (ln > 0.0) {
160      float vh = max(0.0, dot(V, H));
                        float hn = max(0.0, dot(H, N));
                        float d = hn * hn;
                        float D = exp((d - 1.0) / (d * roughness)) / (pi * roughness * d * d);
                        float F;
165      {
                            float c = vh;
                            float g = sqrt(index * index + c * c - 1.0);
                            float a0 = g - c;
                            float a1 = g + c;
170      float a2 = a1 * c - 1.0; a2 *= a2;
                            float a3 = a0 * c - 1.0; a3 *= a3;
                            float a4 = 1.0 + a2 / a3;
                            a0 *= a0;
                            a1 *= a1;
175      F = 0.5 * a0 / a1 * a4;
                        }
                        float g = 2.0 * hn / vh;
                        float G = min(1.0, min(g * vn, g * ln));
                        k = D * F * G / (4.0 * vn * ln);
180      }
                        diff += diffuse * ln;
                        spec += specular * k;
                    }
                }
            }
185      colour = vec4(clamp(q * 2.0 * material * diff + 2.0 * alpha * vec3(spec),
                ↵ vec3(0.0), vec3(1.0)), alpha);
        }
    }
    );

190  static const char *median_vertex = R"(

```

```

#version 410 core
out vec2 c;
void main(void) {
    switch (gl_VertexID) {
195     case 0:
        c = vec2(0.0, 0.0);
        gl_Position = vec4(-1.0, -1.0, 0.0, 1.0);
        break;
        case 1:
200     c = vec2(1.0, 0.0);
        gl_Position = vec4( 1.0, -1.0, 0.0, 1.0);
        break;
        case 2:
        c = vec2(0.0, 1.0);
205     gl_Position = vec4(-1.0,  1.0, 0.0, 1.0);
        break;
        case 3:
        c = vec2(1.0, 1.0);
210     gl_Position = vec4( 1.0,  1.0, 0.0, 1.0);
        break;
    }
}
)";

215 // bubble sort
static const char *median_fragment = R"(
#version 410 core
layout(location = 0, index = 0) out vec4 colour;
uniform sampler2D t;
220 uniform vec2 d;
uniform bool median;
in vec2 c;
void main(void) {
    vec3 v;
225     if (median) {
        vec3 a[25];
        int k = 0;
        for (int i = 0; i < 5; ++i) {
            float x = float(i - 2);
230             for (int j = 0; j < 5; ++j) {
                float y = float(j - 2);
                a[k++] = texture(t, c + d * vec2(x, y)).xyz;
            }
        }
235     for (int p = 0; p < 24; ++p) {
        for (int i = 0; i < 24; ++i) {
            int j = i + 1;
            vec3 mi = min(a[i], a[j]);
            vec3 ma = max(a[i], a[j]);
240             a[i] = mi;
            a[j] = ma;
        }
    }
    v = a[12];
245 } else {
    v = texture(t, c).xyz;
}
}

```

```

    colour = vec4(v, 1.0);
}
250  )”;
```

```

// <http://pippin.gimp.org/a_dither/>
static const char *rgb2yuv_fragment = R”(
#version 410 core
255  layout(location = 0, index = 0) out vec4 colour;
    uniform sampler2D t;
    uniform ivec2 size;
    uniform bool dithering;
    uniform bool deep;
260  uniform bool chromass;
    in vec2 c;
    float dither(ivec3 p) {
        int m = (p.x * 125 + p.y * 81 + p.z * 67) & 255;
        return float(m) / 255.0;
265  }
    float nonlinear(float l) {
        l = clamp(l, 0.0, 1.0);
        if (l >= 0.018) {
            return 1.099 * pow(l, 0.45) - 0.099;
270  } else {
            return 4.500 * l;
        }
    }
    vec3 nonlinear(vec3 l) {
275  return vec3(nonlinear(l.x), nonlinear(l.y), nonlinear(l.z));
    }
    void main(void) {
        int k = int(floor(c.x * float(size.x))) + size.x * int(floor(c.y * float(
            ↪ chromass ? size.y + size.y/2 : size.y * 3)));
        int channel = 0;
280  if (k >= size.x * size.y) {
            k -= size.x * size.y;
            channel++;
            if (chromass) {
                if (k >= (size.x / 2) * (size.y / 2)) {
285  k -= (size.x / 2) * (size.y / 2);
                    channel++;
                }
            } else {
                if (k >= (size.x) * (size.y)) {
290  k -= (size.x) * (size.y);
                    channel++;
                }
            }
        }
295  if (channel == 0) {
        vec3 cspace = vec3(0.2126, 0.7152, 0.0722);
        ivec2 ij = ivec2(mod(k, size.x), k / size.x);
        vec3 rgb = texelFetch(t, ij, 0).rgb;
        rgb = nonlinear(rgb);
300  float y = dot(cspace, rgb);
        y = 219.0 * y + 16.0;
        if (deep) { y *= 256.0; }
        if (dithering) {
```

```

    y += dither(ivec3(ij, channel));
305 } else {
    y += 0.5;
}
float q;
if (deep) {
310 q = clamp(floor(y + 0.5), 16.0 * 256.0, 236.0 * 256.0 - 1.0) / 65535.0;
} else {
    q = clamp(floor(y + 0.5), 16.0, 235.0) / 255.0;
}
colour = vec4(q);
315 } else {
    vec3 cspace = channel == 1
        ? vec3(-0.2126, -0.7152, 0.9278) / 1.8556
        : vec3(0.7874, -0.7152, -0.0722) / 1.5748;
    ivec2 ij = chromass
320 ? ivec2(mod(k, size.x / 2), k / (size.x / 2))
    : ivec2(mod(k, size.x), k / (size.x));
    vec3 rgb;
    if (chromass) {
        rgb
325 = texelFetch(t, 2 * ij + ivec2(0, 0), 0).rgb
        + texelFetch(t, 2 * ij + ivec2(0, 1), 0).rgb
        + texelFetch(t, 2 * ij + ivec2(1, 0), 0).rgb
        + texelFetch(t, 2 * ij + ivec2(1, 1), 0).rgb;
        rgb *= 0.25;
330 } else {
        rgb = texelFetch(t, ij, 0).rgb;
    }
    rgb = nonlinear(rgb);
    float y = dot(cspace, rgb);
335 y = 224.0 * y + 128.0;
    if (deep) { y *= 256.0; }
    if (dithering) {
        y += dither(ivec3(ij, channel));
    } else {
340 y += 0.5;
    }
    float q;
    if (deep) {
        q = clamp(floor(y), 16.0 * 256.0, 241.0 * 256.0 - 1.0) / 65535.0;
345 } else {
        q = clamp(floor(y), 16.0, 240.0) / 255.0;
    }
    colour = vec4(q);
}
350 }
)";

static bool compile_shader(GLuint program, GLenum type, const char *src) {
    GLuint s = glCreateShader(type);
355 glShaderSource(s, 1, &src, 0);
    glCompileShader(s);
    GLint ok = 0;
    glGetShaderiv(s, GL_COMPILE_STATUS, &ok);
    if (ok) {
360 glAttachShader(program, s);

```



```

        glDeleteShader(s);
    } else {
        char buf[1024];
        buf[0] = 0;
365     glGetShaderInfoLog(s, 1024, 0, buf);
        fprintf(stderr, "%s\n", buf);
    }
    return ok;
}
370
static bool link_program(GLuint p) {
    glLinkProgram(p);
    GLint ok = 0;
    glGetProgramiv(p, GL_LINK_STATUS, &ok);
375     if (! ok) {
        char buf[1024];
        buf[0] = 0;
        glGetProgramInfoLog(p, 1024, 0, buf);
        fprintf(stderr, "%s\n", buf);
380     }
    return ok;
}

static GLuint compile_program_morph() {
385     GLuint p = glCreateProgram();
    compile_shader(p, GL_VERTEX_SHADER, morph_vertex);
    compile_shader(p, GL_GEOMETRY_SHADER, morph_geometry);
    compile_shader(p, GL_FRAGMENT_SHADER, morph_fragment);
    link_program(p);
390     return p;
}

static GLuint compile_program_median() {
    GLuint p = glCreateProgram();
395     compile_shader(p, GL_VERTEX_SHADER, median_vertex);
    compile_shader(p, GL_FRAGMENT_SHADER, median_fragment);
    link_program(p);
    return p;
}
400

static GLuint compile_program_rgb2yuv() {
    GLuint p = glCreateProgram();
    compile_shader(p, GL_VERTEX_SHADER, median_vertex);
    compile_shader(p, GL_FRAGMENT_SHADER, rgb2yuv_fragment);
405     link_program(p);
    return p;
}

static void *read_file(const char *filename, size_t *size) {
410     FILE *f = fopen(filename, "rb");
    fseek(f, 0, SEEK_END);
    *size = ftell(f);
    fseek(f, 0, SEEK_SET);
    void *buf = malloc(*size);
415     fread(buf, *size, 1, f);
    fclose(f);
    return buf;
}

```

```

}

420 extern int main(int argc, char **argv) {
    (void) argc;
    (void) argv;

    int FPS = 30;
425 int width = 1920;
    int height = 1080;
    int samples = 8;
    double speedup = 1;
    bool excerpt = false;
430 double excerpt_start = 0.25; // percentage
    double excerpt_length = 40; // seconds
    bool filter = true;
    bool median = false;
    bool deep = true;
435 bool deepio = false;
    bool record = true;
    bool rgb = true;
    bool chromass = false;
    bool dithering = false;
440 int pipeline = 4;

    glfwInit();
    glfwWindowHint(GLFW_CLIENT_API, GLFW_OPENGL_API);
    glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, 4);
445 glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, 4);
    glfwWindowHint(GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE);
    glfwWindowHint(GLFW_OPENGL_FORWARD_COMPAT, GL_TRUE);
    glfwWindowHint(GLFW_RESIZABLE, GL_FALSE);
    GLFWwindow *window = glfwCreateWindow(width, height, "Umstülpung", 0, 0);

450 glfwMakeContextCurrent(window);
    glewExperimental = GL_TRUE;
    glewInit();
    glGetError();

455 GLuint mp = compile_program_median();
    glUseProgram(mp);
    GLint mu_d = glGetUniformLocation(mp, "d");
    GLint mu_median = glGetUniformLocation(mp, "median");
460 glUniform2f(mu_d, 1.0 / width, 1.0 / height);
    glUniform1i(mu_median, median ? 1 : 0);

    GLuint yp = compile_program_rgb2yuv();
    glUseProgram(yp);
465 GLint yu_t = glGetUniformLocation(yp, "t");
    GLint yu_size = glGetUniformLocation(yp, "size");
    GLint yu_dithering = glGetUniformLocation(yp, "dithering");
    GLint yu_deep = glGetUniformLocation(yp, "deep");
    GLint yu_chromass = glGetUniformLocation(yp, "chromass");
470 glUniform1i(yu_t, filter ? 1 : 0);
    glUniform2i(yu_size, width, height);
    glUniform1i(yu_dithering, dithering ? 1 : 0);
    glUniform1i(yu_deep, deepio ? 1 : 0);
    glUniform1i(yu_chromass, chromass ? 1 : 0);

```

```

475     GLuint p = compile_program_morph();
        glUseProgram(p);
        GLint u_morph = glGetUniformLocation(p, "morph");
        GLint u_projectionMatrix = glGetUniformLocation(p, "projectionMatrix");
480     GLint u_modelMatrix = glGetUniformLocation(p, "modelMatrix");
        GLint u_normalMatrix = glGetUniformLocation(p, "normalMatrix");
        GLint u_scale = glGetUniformLocation(p, "scale");
        GLint u_eye = glGetUniformLocation(p, "eye");
        GLint u_material = glGetUniformLocation(p, "material");
485     GLint u_ambient = glGetUniformLocation(p, "ambient");
        GLint u_diffuse = glGetUniformLocation(p, "diffuse");
        GLint u_specular = glGetUniformLocation(p, "specular");
        GLint u_index = glGetUniformLocation(p, "index");
        GLint u_roughness = glGetUniformLocation(p, "roughness");
490     GLint u_alpha = glGetUniformLocation(p, "alpha");

        float aspect = width / (float) height;
        float size = 7.0f / 4.0f * 10.0 / 9.0;
        glm::mat4 projectionMatrix(glm::ortho(-aspect * size, aspect * size, -size, ↵
            ↵ size, 0.1f, 10.0f));
495     glUniformMatrix4fv(u_projectionMatrix, 1, GL_FALSE, &projectionMatrix[0][0]);

        glUniform3f(u_eye, 0, 0, 0);

500     // <http://gemologyproject.com/wiki/index.php?title=Obsidian>
        glUniform3f(u_material, 0, 0, 0);
        glUniform1f(u_ambient, 0.01);
        glUniform1f(u_diffuse, 0.2);
        glUniform1f(u_specular, 0.8);
505     glUniform1f(u_index, 1.5);
        glUniform1f(u_roughness, 0.15 * 0.15);
        glUniform1f(u_alpha, 1);

        GLuint texy;
510     glGenTextures(1, &texy);
        glBindTexture(GL_TEXTURE_2D, texy);
        glTexImage2D(GL_TEXTURE_2D, 0, deepio ? GL_R16 : GL_R8, width, chromass ? (↵
            ↵ height + height/2) : (height * 3), 0, GL_RED, deepio ? GL_UNSIGNED_SHORT↵
            ↵ : GL_UNSIGNED_BYTE, 0);
        glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
        glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
515     glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
        glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
        GLuint fboy;
        glGenFramebuffers(1, &fboy);
        glBindFramebuffer(GL_FRAMEBUFFER, fboy);
520     glFramebufferTexture2D(GL_FRAMEBUFFER, GL_COLOR_ATTACHMENT0, GL_TEXTURE_2D, ↵
            ↵ texy, 0);
        int bytes = chromass ? width * (height + height / 2) : width * (height * 3);
        if (deepio) { bytes *= 2; }
        GLsync syncs[pipeline];
        GLuint pbo[pipeline];
525     glGenBuffers(pipeline, &pbo[0]);
        for (int i = 0; i < pipeline; ++i) {
            glBindBuffer(GL_PIXEL_PACK_BUFFER, pbo[i]);

```

```

    glBufferStorage(GL_PIXEL_PACK_BUFFER, bytes, 0, GL_MAP_READ_BIT);
}
530
GLuint texd;
glGenTextures(1, &texd);
glBindTexture(GL_TEXTURE_2D_MULTISAMPLE, texd);
glTexImage2DMultisample(GL_TEXTURE_2D_MULTISAMPLE, samples, GL_DEPTH_COMPONENT,
    ↪ , width, height, false);
535
GLuint tex;
glGenTextures(1, &tex);
glBindTexture(GL_TEXTURE_2D_MULTISAMPLE, tex);
glTexImage2DMultisample(GL_TEXTURE_2D_MULTISAMPLE, samples, deep ? GL_RGBA16 :
    ↪ GL_RGBA8, width, height, false);
GLuint fbo;
540
glGenFramebuffers(1, &fbo);
glBindFramebuffer(GL_FRAMEBUFFER, fbo);
glFramebufferTexture2D(GL_FRAMEBUFFER, GL_COLOR_ATTACHMENT0,
    ↪ GL_TEXTURE_2D_MULTISAMPLE, tex, 0);
glFramebufferTexture2D(GL_FRAMEBUFFER, GL_DEPTH_ATTACHMENT,
    ↪ GL_TEXTURE_2D_MULTISAMPLE, texd, 0);

545
GLuint tex2;
glGenTextures(1, &tex2);
glBindTexture(GL_TEXTURE_2D, tex2);
glTexImage2D(GL_TEXTURE_2D, 0, deep ? GL_RGBA16 : GL_RGBA8, width, height, 0,
    ↪ GL_RGBA, GL_UNSIGNED_BYTE, 0);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
550
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
GLuint fbo2;
glGenFramebuffers(1, &fbo2);
555
glBindFramebuffer(GL_FRAMEBUFFER, fbo2);
glFramebufferTexture2D(GL_FRAMEBUFFER, GL_COLOR_ATTACHMENT0, GL_TEXTURE_2D,
    ↪ tex2, 0);

GLuint vao;
glGenVertexArrays(1, &vao);
560
glBindVertexArray(vao);
GLuint ebo;
glGenBuffers(1, &ebo);
glBindBuffer(GL_ELEMENT_ARRAY_BUFFER, ebo);
size_t cuboloid_indices_size = 0;
565
GLuint *cuboloid_indices = (GLuint *) read_file("cuboloid.u32", &
    ↪ cuboloid_indices_size);
glBufferData(GL_ELEMENT_ARRAY_BUFFER, cuboloid_indices_size, cuboloid_indices,
    ↪ GL_STATIC_DRAW);
free(cuboloid_indices);

GLuint vbo;
570
glGenBuffers(1, &vbo);
glBindBuffer(GL_ARRAY_BUFFER, vbo);
size_t cuboloid_vertices_size = 0;
GLfloat *cuboloid_vertices = (GLfloat *) read_file("cuboloid.f32", &
    ↪ cuboloid_vertices_size);
glBufferData(GL_ARRAY_BUFFER, cuboloid_vertices_size, cuboloid_vertices,
    ↪ GL_STATIC_DRAW);

```

```

575     free(cuboloid_vertices);

    glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 6 * sizeof(GLfloat), 0);
    glVertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, 6 * sizeof(GLfloat), ((
        ↵ unsigned char *) 0) + 3 * sizeof(GLfloat));
    glEnableVertexAttribArray(0);
580    glEnableVertexAttribArray(1);

    glActiveTexture(GL_TEXTURE1);
    GLuint vaom;
    glGenVertexArrays(1, &vaom);
585    GLuint texm;
    glGenTextures(1, &texm);
    glBindTexture(GL_TEXTURE_2D, texm);
    glTexImage2D(GL_TEXTURE_2D, 0, deep ? GL_RGBA16 : GL_RGBA8, width, height, 0, ↵
        ↵ GL_RGBA, GL_UNSIGNED_BYTE, 0);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
590    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP_TO_EDGE);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP_TO_EDGE);
    GLuint fbom;
    glGenFramebuffers(1, &fbom);
595    glBindFramebuffer(GL_FRAMEBUFFER, fbom);
    glFramebufferTexture2D(GL_FRAMEBUFFER, GL_COLOR_ATTACHMENT0, GL_TEXTURE_2D, ↵
        ↵ texm, 0);
    glActiveTexture(GL_TEXTURE0);

    if (deepio && rgb) {
600        glPixelStorei(GL_PACK_SWAP_BYTES, GL_TRUE);
    }

    // average earth-sun light minutes 8.317
    double t0 = 8.317 * 60 / speedup;
605

    // whole numbers of frames for looping
    int frames = round(2 * t0 * FPS);
    t0 = frames / (2.0 * FPS);
    fprintf(stderr, "%d frames = 2 * %f seconds * %d frames per second\n", frames, ↵
        ↵ t0, FPS);
610

    // excerpt
    int frame0 = 0;
    int frame1 = frames / 2;
    if (excerpt) {
615        frame0 = excerpt_start * frames;
        frame1 = excerpt_length * FPS;
    }

    int frame = 0;
620    while (! glfwWindowShouldClose(window)) {
        double f = (frame + frame0) % frames;
        double x = f / (double) frames;
        double z = x;
        double w = x;
625        float h = fmod(9 * w, 1);
        float s = w < 0.25 ? 4 * w : w > 0.75 ? 4 * (1 - w) : 1;
        s = 0.5 - 0.5 * cos(pi * s);
    }

```

```

float v = w < 0.5 ? 1 - 2 * w : 2 * w - 1;
float r, g, b;
630 hsv2rgb(h, s, v, &r, &g, &b);
glClearColor(1 - r, 1 - g, 1 - b, 1);

glEnable(GL_DEPTH_TEST);
if (filter) {
635 glBindFramebuffer(GL_FRAMEBUFFER, fbo);
} else if (record) {
glBindFramebuffer(GL_FRAMEBUFFER, fbo2);
} else {
640 glBindFramebuffer(GL_FRAMEBUFFER, 0);
}

glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

glBindVertexArray(vao);
645 glUseProgram(p);
glUniform1f(u_scale, 1);
glUniform3f(u_material, r, g, b);
glUniform1f(u_morph, 0.5 + 0.5 * cos(twopi * x));
glm::mat4 rot1(glm::rotate(glm::mat4(1.0f), float(twopi * 2 * z), glm::vec3(
    ↪ (0.0f, 1.0f, 0.0f))));
650 glm::mat4 rot2(glm::rotate(glm::mat4(1.0f), float(twopi * 3 * z), glm::vec3(
    ↪ (0.0f, 0.0f, 1.0f))));
glm::mat4 rot3(glm::rotate(glm::mat4(1.0f), float(twopi * 4 * z), glm::vec3(
    ↪ (1.0f, 0.0f, 0.0f))));
glm::mat4 modelMatrix1(glm::translate(glm::mat4(1.0f), glm::vec3(0.0f, 0.0f,
    ↪ -3.0f))));
glm::mat4 modelMatrix(modelMatrix1 * rot1 * rot2 * rot3);
glUniformMatrix4fv(u_modelMatrix, 1, GL_FALSE, &modelMatrix[0][0]);
655 glm::mat3 normalMatrix(glm::inverse(glm::transpose(glm::mat3(modelMatrix))))
    ↪ ;
glUniformMatrix3fv(u_normalMatrix, 1, GL_FALSE, &normalMatrix[0][0]);

glDrawElements(GL_TRIANGLES, cuboloid_indices_size / sizeof(GLuint),
    ↪ GL_UNSIGNED_INT, 0);

660 if (filter) {
glBindFramebuffer(GL_FRAMEBUFFER, 0);
}
glDisable(GL_DEPTH_TEST);

665 if (filter) {
glBindFramebuffer(GL_DRAW_FRAMEBUFFER, fbo2);
glBindFramebuffer(GL_READ_FRAMEBUFFER, fbo);
glBlitFramebuffer(0, 0, width, height, 0, 0, width, height,
    ↪ GL_COLOR_BUFFER_BIT, GL_NEAREST);
glBindFramebuffer(GL_DRAW_FRAMEBUFFER, 0);
670 glBindFramebuffer(GL_READ_FRAMEBUFFER, 0);
if (record) {
glBindFramebuffer(GL_FRAMEBUFFER, fbom);
} else {
glBindFramebuffer(GL_FRAMEBUFFER, 0);
675 }
glBindVertexArray(vaom);
glUseProgram(mp);

```

```

        glDrawArrays(GL_TRIANGLE_STRIP, 0, 4);
    }
680
    if (record) {
        glBindFramebuffer(GL_FRAMEBUFFER, 0);
        glBindFramebuffer(GL_DRAW_FRAMEBUFFER, 0);
        glBindFramebuffer(GL_READ_FRAMEBUFFER, filter ? fbom : fbo2);
685        glBlitFramebuffer(0, 0, width, height, 0, 0, width, height, ↵
            ↵ GL_COLOR_BUFFER_BIT, GL_NEAREST);

        int k = frame % pipeline;
        glBindBuffer(GL_PIXEL_PACK_BUFFER, pbo[k]);
        GLsync s = glFenceSync(GL_SYNC_GPU_COMMANDS_COMPLETE, 0);
690        if (frame - pipeline >= 0) {
            glWaitSync(syncs[k], 0, GL_TIMEOUT_IGNORED);
            void *buf = glMapBufferRange(GL_PIXEL_PACK_BUFFER, 0, bytes, ↵
                ↵ GL_MAP_READ_BIT);
            if (rgb) {
                fprintf(stdout, "P6\n%d %d\n%d\n", width, height, deepio ? 65535 : ↵
                    ↵ 255);
695            }
            fwrite(buf, bytes, 1, stdout);
            glUnmapBuffer(GL_PIXEL_PACK_BUFFER);
        }
        syncs[k] = s;
700        if (rgb) {
            glReadPixels(0, 0, width, height, GL_RGB, deepio ? GL_UNSIGNED_SHORT : ↵
                ↵ GL_UNSIGNED_BYTE, 0);
        } else {
            glViewport(0, 0, width, chromass ? (height + height / 2) : (height * 3))↵
                ↵ ;
            glBindFramebuffer(GL_FRAMEBUFFER, fboy);
705            glBindVertexArray(vaom);
            glUseProgram(yp);
            glDrawArrays(GL_TRIANGLE_STRIP, 0, 4);
            glReadPixels(0, 0, width, chromass ? (height + height / 2) : (height * ↵
                ↵ 3), GL_RED, deepio ? GL_UNSIGNED_SHORT : GL_UNSIGNED_BYTE, 0);
            glBindFramebuffer(GL_FRAMEBUFFER, 0);
710            glViewport(0, 0, width, height);
        }
    }

    { int e = glGetError(); if (e) { fprintf(stderr, "%d: %d\n", __LINE__, e); }↵
        ↵ }
715    glfwSwapBuffers(window);

    frame += 1;
    if (record && frame == frame1 + pipeline) {
        break;
720    }
    glfwPollEvents();
}

return 0;
725 }
```

7 chapter-2/oloid.gnuplot

```

set terminal pngcairo enhanced size 1024,1024
unset key
unset tics
unset border
5  unset colorbox
   set bmargin at screen 0
   set lmargin at screen 0
   set rmargin at screen 1
   set tmargin at screen 1
10 set parametric
   set samples 100
   set isosamples 50,25
   set palette rgb 4,5,6
   #set style line 1 lc rgb '#000000' lt 1 lw 1
15 set pm3d depthorder
   set hidden3d
   set style fill transparent solid 0.5
   set view 30,180
   max(a,b)=a>b?a:b
20 s(t) = 2*pi/3 * sgn(t) *(3*abs(t)-2*abs(t)**1.5)
   ax(t)=sin(s(t))
   ay(t)=-cos(s(t))
   az(t)=0
   bx(t)=0
25 by(t)=1/(1+cos(s(t)))
   bz(t)=sqrt(1+2*cos(s(t)))/(1+cos(s(t)))
   l(u,v)=sqrt(ox(u,v)**2 + (oy(u,v))**2 + oz(u,v)**2)
   ox(u,v)=ax(u)*(1-v)+v*bx(u)
   oy(u,v)=ay(u)*(1-v)+v*by(u) - 0.5
30 oz(u,v)=az(u)*(1-v)+v*bz(u)
   px(u,v)=ox(u,v)/l(u,v)
   py(u,v)=(oy(u,v))/l(u,v)
   pz(u,v)=oz(u,v)/l(u,v)
   m(u,v)=max(max(abs(px(u,v)),abs(py(u,v))),abs(pz(u,v)))* (8.0/3.052418468) ↵
   ↵ ** (1.0/3.0)
35 qx(u,v)=px(u,v)/m(u,v)
   qy(u,v)=py(u,v)/m(u,v)
   qz(u,v)=pz(u,v)/m(u,v)
   rx(u,v)=ox(u,v)*(1-w)+w*qx(u,v)
   ry(u,v)=oy(u,v)*(1-w)+w*qy(u,v)
40 rz(u,v)=oz(u,v)*(1-w)+w*qz(u,v)
   x(u,v)=rx(u,v)*cos(phi)-sin(phi)*ry(u,v)
   y(u,v)=rx(u,v)*sin(phi)+cos(phi)*ry(u,v)
   z(u,v)=rz(u,v)
   set xrange [-1.6:1.6]
45 set yrange [-1.6:1.6]
   set zrange [-1.6:1.6]
   phi=-pi/4
   theta=atan2(1,sqrt(2))
   do for [f=0:100] {
50   set output sprintf('oloid-%03d.png',f)
   w=f/100.0
   splot [-1:1][-0:1] \
     x(u,v),y(u,v)*cos(theta)-sin(theta)*z(u,v),y(u,v)*sin(theta)+cos(theta)*z(u,v) ↵
     ↵ w pm3d, \

```



```

55  x(u,v),y(u,v)*cos(theta)+sin(theta)*z(u,v),y(u,v)*sin(theta)-cos(theta)*z(u,v) ↵
    ↵ w pm3d
    }

```

8 LICENSE

"Self-portrait at the auto-da-fe" popobobo oddodood, Claude Heiland-Allen 2013
 Copyleft: This is a free work, you can copy, distribute, and modify it
 under the terms of the Free Art License <http://artlibre.org/licence/lal/en/>

9 prototype/anaglyph-help.pd

```

#N canvas 3 63 592 597 10;
#X obj 13 120 anaglyph;
#X obj 13 146 translateXYZ 0 0 0;
#X obj 17 398 t a a;
5 #X obj 14 169 colorRGB 1 1 1;
#X obj 188 445 throw~ la;
#X obj 256 443 throw~ ra;
#N canvas 0 50 372 399 \ $0-lightnings 0;
#X obj 35 22 inlet;
10 #X obj 228 16 inlet;
#X obj 49 161 spigot;
#X obj 132 112 delay;
#X obj 136 21 loadbang;
#X msg 136 42 1;
15 #X obj 49 182 random 600;
#X obj 49 203 - 300;
#X obj 41 255 spigot;
#X obj 92 136 < 100;
#X obj 106 279 s go;
20 #X msg 97 208 0;
#X obj 106 257 sel 0;
#X obj 234 168 spigot;
#X obj 209 262 spigot;
#X obj 285 284 s go;
25 #X msg 276 213 0;
#X obj 285 262 sel 0;
#X obj 277 143 > 900;
#X obj 285 241 pipe f 2000;
#X obj 234 189 random 120;
30 #X obj 234 210 - 60;
#X obj 106 236 pipe f 2000;
#X obj 131 69 metro 4000;
#X obj 132 91 random 3750;
#X obj 26 102 t a a a;
35 #X obj 17 352 GEMglClearColor;
#X obj 39 323 t f f f;
#X obj 41 280 lightning;
#X obj 209 283 lightning;
#X connect 0 0 25 0;
40 #X connect 1 0 18 0;
#X connect 1 0 9 0;
#X connect 2 0 6 0;
#X connect 2 0 11 0;
#X connect 3 0 2 0;

```

```

45  #X connect 3 0 13 0;
    #X connect 4 0 5 0;
    #X connect 5 0 23 0;
    #X connect 6 0 7 0;
    #X connect 7 0 28 1;
50  #X connect 8 0 28 0;
    #X connect 9 0 2 1;
    #X connect 9 0 8 1;
    #X connect 11 0 22 0;
    #X connect 12 0 10 0;
55  #X connect 13 0 16 0;
    #X connect 13 0 20 0;
    #X connect 14 0 29 0;
    #X connect 16 0 19 0;
    #X connect 17 0 15 0;
60  #X connect 18 0 13 1;
    #X connect 18 0 14 1;
    #X connect 19 0 17 0;
    #X connect 20 0 21 0;
    #X connect 21 0 29 1;
65  #X connect 22 0 12 0;
    #X connect 23 0 24 0;
    #X connect 24 0 3 0;
    #X connect 25 0 26 0;
    #X connect 25 1 8 0;
70  #X connect 25 2 14 0;
    #X connect 27 0 26 1;
    #X connect 27 1 26 2;
    #X connect 27 2 26 3;
    #X connect 28 0 27 0;
75  #X connect 29 0 27 0;
    #X restore 66 435 pd \ $0-lightnings;
    #X obj 17 377 t a a;
    #X obj 15 333 t a a;
    #N canvas 0 50 522 510 \ $0-violent-orb 0;
80  #X obj 20 17 inlet;
    #X obj 188 2 inlet;
    #X obj 265 357 outlet~;
    #X obj 320 359 outlet~;
    #X obj 194 257 clip 0 1;
85  #X obj 231 190 +~ 0.25;
    #X obj 231 211 wrap~;
    #X obj 368 188 +~ 0.25;
    #X obj 368 209 wrap~;
    #X obj 267 315 samphold~;
90  #X obj 333 315 samphold~;
    #X obj 267 336 *~;
    #X obj 333 336 *~;
    #X obj 271 135 phasor~ 60;
    #X obj 286 42 snapshot~;
95  #X obj 286 94 + 6;
    #X obj 286 63 * 0.125;
    #X obj 368 167 phasor~;
    #X obj 267 294 cos~;
    #X obj 333 294 cos~;
100 #X obj 319 202 +~ 0.25;
    #X obj 319 223 wrap~;

```

```

#X obj 294 154 *~ 3;
#X obj 294 175 wrap~;
#X obj 326 245 cos~;
105 #X obj 276 246 cos~;
#X obj 370 93 osc~;
#X obj 369 64 +~ 3;
#X obj 373 33 osc~ 0.1;
#X obj 369 114 *~ 300;
110 #X obj 276 267 *~;
#X obj 326 266 *~;
#X obj 369 135 +~ 3000;
#X obj 20 38 t a b;
#X obj 88 89 with-lighting;
115 #X obj 87 68 separator;
#X obj 48 258 sphere 6;
#X obj 48 196 translateXYZ 0 8 0;
#X obj 65 115 f 0;
#X obj 124 131 mod 360;
120 #X obj 49 238 rotateXYZ 90 0 0;
#X obj 48 217 rotateXYZ;
#X obj 124 111 + 3;
#X obj 97 13 inlet;
#X obj 188 23 t f f;
125 #X obj 188 44 *;
#X obj 190 91 swap 900;
#X obj 190 112 /;
#X obj 437 37 inlet;
#X obj 437 58 * 2;
130 #X obj 437 79 * 3.14159;
#X obj 437 100 sin;
#X obj 437 121 - 1;
#X obj 437 142 / -2;
#X obj 437 184 + 3000;
135 #X obj 437 163 * 1000;
#X connect 0 0 33 0;
#X connect 1 0 44 0;
#X connect 4 0 11 1;
#X connect 4 0 12 1;
140 #X connect 5 0 6 0;
#X connect 6 0 19 0;
#X connect 7 0 8 0;
#X connect 8 0 10 1;
#X connect 9 0 11 0;
145 #X connect 10 0 12 0;
#X connect 11 0 2 0;
#X connect 12 0 3 0;
#X connect 13 0 5 0;
#X connect 13 0 22 0;
150 #X connect 13 0 18 0;
#X connect 14 0 16 0;
#X connect 15 0 36 1;
#X connect 16 0 15 0;
#X connect 17 0 7 0;
155 #X connect 17 0 9 1;
#X connect 18 0 9 0;
#X connect 19 0 10 0;
#X connect 20 0 21 0;

```

```
#X connect 21 0 24 0;
160 #X connect 22 0 23 0;
#X connect 23 0 20 0;
#X connect 23 0 25 0;
#X connect 24 0 31 0;
#X connect 25 0 30 0;
165 #X connect 26 0 14 0;
#X connect 26 0 29 0;
#X connect 26 0 30 1;
#X connect 26 0 31 1;
#X connect 27 0 26 0;
170 #X connect 28 0 27 0;
#X connect 29 0 32 0;
#X connect 30 0 18 0;
#X connect 31 0 19 0;
#X connect 32 0 17 0;
175 #X connect 33 0 35 0;
#X connect 33 1 14 0;
#X connect 34 0 37 0;
#X connect 35 0 34 0;
#X connect 37 0 41 0;
180 #X connect 38 0 41 2;
#X connect 38 0 42 0;
#X connect 39 0 38 1;
#X connect 40 0 36 0;
#X connect 41 0 40 0;
185 #X connect 42 0 39 0;
#X connect 43 0 38 0;
#X connect 44 0 45 0;
#X connect 44 1 45 1;
#X connect 45 0 46 0;
190 #X connect 46 0 47 0;
#X connect 46 1 47 1;
#X connect 47 0 4 0;
#X connect 48 0 49 0;
#X connect 49 0 50 0;
195 #X connect 50 0 51 0;
#X connect 51 0 52 0;
#X connect 52 0 53 0;
#X connect 53 0 55 0;
#X connect 54 0 32 1;
200 #X connect 55 0 54 0;
#X restore 65 358 pd \ $0-violent-orb;
#N canvas 554 50 191 128 \ $0-mountain-range 0;
#X obj 25 15 inlet;
#X obj 25 42 mountainss;
205 #X connect 0 0 1 0;
#X restore 56 387 pd \ $0-mountain-range;
#X obj 15 353 t a a;
#N canvas 559 109 710 865 \ $0-bouncers 0;
#X obj 21 68 separator;
210 #X obj 253 393 / 180;
#X obj 253 414 * 3.14159;
#X obj 253 435 cos;
#X obj 238 471 abs;
#X obj 253 352 +;
215 #X obj 23 192 separator;
```

```
#X obj 254 372 t f f;
#X obj 238 492 * 8;
#X obj 238 513 - 5;
#X obj 245 178 t f f;
220 #X obj 280 252 *;
#X obj 306 288 t f f;
#X obj 306 309 *;
#X obj 306 351 /;
#X obj 279 227 + 1;
225 #X obj 306 330 swap 16;
#X obj 281 203 mod 3;
#X obj 253 316 * 144;
#X obj 221 603 outlet~;
#X obj 282 605 outlet~;
230 #X obj 425 23 inlet;
#X obj 21 32 inlet;
#X obj 320 46 inlet;
#X obj 21 89 with-lighting;
#X obj 426 63 * 20;
235 #X obj 427 101 * 60;
#X obj 428 175 mod 360;
#X obj 427 118 * 25;
#X obj 507 195 * 6;
#X obj 507 216 wrap;
240 #X obj 507 237 - 0.5;
#X obj 507 258 * 120;
#X obj 425 44 t f f;
#X obj 427 82 t f f;
#X obj 479 169 * 2;
245 #X obj 479 190 - 1;
#X obj 479 211 *;
#X obj 479 148 expr $f1 < 10;
#X obj 21 158 repeat-gemlist 4;
#X obj 84 331 list split 1;
250 #X obj 24 388 translate 1 0 0 0;
#X obj 223 567 ball-sounds;
#X obj 102 191 t f f;
#X obj 167 187 +;
#X obj 427 139 * 3;
255 #X msg 130 353 \$1 \$2 0;
#X obj 88 289 pack f f f f f;
#X obj 167 166 * 8;
#X obj 167 208 - 12;
#X obj 465 372 < 0.5;
260 #X obj 465 456 * 200;
#X obj 399 374 - 700;
#X obj 20 137 translateXYZ 0 0 -650;
#X obj 30 439 spigot 1;
#X obj 93 441 spigot 0;
265 #X obj 28 501 sphere 1 20;
#X obj 113 536 cube;
#X obj 93 462 rotateXYZ 45 30 45;
#X obj 153 443 ==;
#X obj 465 414 + 3;
270 #X obj 465 435 / 4;
#X obj 465 393 change -1;
#X obj 399 395 * -6;
```

```
#X connect 0 0 24 0;
#X connect 1 0 2 0;
275 #X connect 2 0 3 0;
#X connect 3 0 4 0;
#X connect 3 0 47 3;
#X connect 4 0 8 0;
#X connect 5 0 7 0;
280 #X connect 6 0 41 0;
#X connect 7 0 1 0;
#X connect 8 0 9 0;
#X connect 9 0 47 2;
#X connect 10 0 18 0;
285 #X connect 10 1 17 0;
#X connect 11 0 5 1;
#X connect 12 0 13 0;
#X connect 12 1 13 1;
#X connect 13 0 16 0;
290 #X connect 14 0 8 1;
#X connect 14 0 47 4;
#X connect 15 0 11 0;
#X connect 15 0 12 0;
#X connect 16 0 14 0;
295 #X connect 16 1 14 1;
#X connect 17 0 15 0;
#X connect 18 0 5 0;
#X connect 21 0 33 0;
#X connect 22 0 0 0;
300 #X connect 23 0 52 0;
#X connect 24 0 53 0;
#X connect 25 0 34 0;
#X connect 26 0 28 0;
#X connect 27 0 11 1;
305 #X connect 28 0 45 0;
#X connect 29 0 30 0;
#X connect 30 0 31 0;
#X connect 31 0 32 0;
#X connect 32 0 37 1;
310 #X connect 33 0 25 0;
#X connect 33 1 29 0;
#X connect 33 1 50 0;
#X connect 34 0 26 0;
#X connect 34 1 38 0;
315 #X connect 35 0 36 0;
#X connect 36 0 37 0;
#X connect 37 0 44 1;
#X connect 38 0 35 0;
#X connect 39 0 6 0;
320 #X connect 39 1 43 0;
#X connect 40 1 46 0;
#X connect 41 0 54 0;
#X connect 41 0 55 0;
#X connect 42 0 19 0;
325 #X connect 42 1 20 0;
#X connect 43 0 47 0;
#X connect 43 1 10 0;
#X connect 43 1 48 0;
#X connect 44 0 49 0;
```

```
330 #X connect 45 0 27 0;
    #X connect 46 0 41 2;
    #X connect 47 0 40 0;
    #X connect 47 0 42 0;
    #X connect 48 0 44 0;
335 #X connect 49 0 47 1;
    #X connect 50 0 54 1;
    #X connect 50 0 59 0;
    #X connect 50 0 62 0;
    #X connect 51 0 42 2;
340 #X connect 52 0 63 0;
    #X connect 53 0 39 0;
    #X connect 54 0 56 0;
    #X connect 55 0 58 0;
    #X connect 58 0 57 0;
345 #X connect 59 0 55 1;
    #X connect 60 0 61 0;
    #X connect 61 0 51 0;
    #X connect 62 0 60 0;
    #X connect 63 0 42 1;
350 #X restore 62 310 pd \ $0-bouncers;
    #N canvas 312 50 697 510 \ $0-worms 0;
    #X obj 110 290 snake;
    #X obj 115 240 snakes;
    #X obj 108 270 gemlist;
355 #X obj 164 207 loadbang;
    #X obj 107 162 with-lighting;
    #X obj 112 204 t b a;
    #X obj 107 140 separator;
    #X obj 99 26 inlet;
360 #X obj 258 32 inlet;
    #X obj 365 27 inlet;
    #X obj 269 483 outlet~;
    #X obj 333 481 outlet~;
    #X msg 175 229 reset 12;
365 #X obj 127 427 scaleXYZ 0.1 0.1 0.01;
    #X obj -1 421 scaleXYZ 0.05 0.05 1;
    #X obj 0 482 cylinder;
    #X obj 168 449 scale 0.2;
    #X obj 168 470 sphere 1 20;
370 #X msg 247 270 wriggle;
    #X msg 195 270 column;
    #X obj 488 102 once 1;
    #X obj 571 103 once 1;
    #X obj 414 102 once 1;
375 #X msg 546 140 1;
    #X msg 467 139 1;
    #X msg 607 140 1;
    #X obj 366 50 * 20;
    #X obj 571 82 moses 20;
380 #X obj 488 81 moses 17;
    #X obj 414 81 moses 6.5;
    #X obj 519 161 - 6.5;
    #X obj 519 203 clip 0 1;
    #X msg 510 240 height \ $1;
385 #X obj 335 50 * 60;
    #X obj 335 71 t f f;
```

```
#X obj 338 92 -;
#X msg 235 151 tick \$1;
#X obj 301 135 clip 0 1;
390 #X msg 164 290 0;
#X obj 519 182 / 3;
#X msg 164 351 0.3;
#X obj 161 310 unpack f f f;
#X obj 207 361 *;
395 #X obj 64 483 cone;
#X obj 207 342 / 8;
#X obj -3 460 spigot 1;
#X obj 62 460 spigot 0;
#X obj 104 439 ==;
400 #X obj 19 323 < 5;
#X obj 19 344 ||;
#X obj 19 286 t f f;
#X obj 52 325 > 15;
#X obj 184 384 t f f f f;
405 #X obj 231 405 * 0.1;
#X obj 83 367 spigot 1;
#X obj 110 333 spigot 0;
#X obj 172 332 ==;
#X obj 129 490 sphere;
410 #X obj 83 388 scale 0.1;
#X obj 338 113 / 2;
#X obj 108 184 translateXYZ 0 -5 -250;
#X obj 269 457 worm-sounds;
#X obj 305 174 - 300;
415 #X obj 310 424 * -1;
#X connect 0 0 14 0;
#X connect 0 1 54 0;
#X connect 0 1 55 0;
#X connect 0 2 16 0;
420 #X connect 0 3 41 0;
#X connect 1 0 2 0;
#X connect 1 1 0 1;
#X connect 1 2 61 0;
#X connect 2 0 0 0;
425 #X connect 3 0 12 0;
#X connect 4 0 60 0;
#X connect 5 0 1 0;
#X connect 5 1 2 1;
#X connect 6 0 4 0;
430 #X connect 7 0 6 0;
#X connect 8 0 62 0;
#X connect 9 0 26 0;
#X connect 12 0 1 1;
#X connect 13 0 57 0;
435 #X connect 14 0 45 0;
#X connect 14 0 46 0;
#X connect 16 0 17 0;
#X connect 18 0 1 1;
#X connect 18 0 61 1;
440 #X connect 19 0 1 1;
#X connect 19 0 61 1;
#X connect 20 0 19 0;
#X connect 20 0 24 0;
```



```

#X connect 20 0 25 0;
445 #X connect 21 0 18 0;
#X connect 21 0 24 0;
#X connect 21 0 23 0;
#X connect 22 0 23 0;
#X connect 22 0 25 0;
450 #X connect 23 0 20 1;
#X connect 24 0 22 1;
#X connect 25 0 21 1;
#X connect 26 0 29 0;
#X connect 26 0 33 0;
455 #X connect 26 0 50 0;
#X connect 27 0 21 0;
#X connect 28 0 20 0;
#X connect 28 0 30 0;
#X connect 28 1 27 0;
460 #X connect 29 0 22 0;
#X connect 29 1 28 0;
#X connect 30 0 39 0;
#X connect 31 0 32 0;
#X connect 31 0 42 1;
465 #X connect 32 0 1 1;
#X connect 33 0 34 0;
#X connect 34 0 35 1;
#X connect 34 1 35 0;
#X connect 35 0 59 0;
470 #X connect 36 0 1 1;
#X connect 37 0 36 0;
#X connect 39 0 31 0;
#X connect 40 0 52 0;
#X connect 41 1 44 0;
475 #X connect 42 0 52 0;
#X connect 44 0 42 0;
#X connect 45 0 15 0;
#X connect 46 0 43 0;
#X connect 47 0 46 1;
480 #X connect 48 0 49 0;
#X connect 49 0 45 1;
#X connect 49 0 47 0;
#X connect 49 0 54 1;
#X connect 49 0 56 0;
485 #X connect 50 0 48 0;
#X connect 50 1 51 0;
#X connect 51 0 49 1;
#X connect 52 0 13 1;
#X connect 52 1 13 2;
490 #X connect 52 2 53 0;
#X connect 53 0 13 3;
#X connect 54 0 58 0;
#X connect 55 0 13 0;
#X connect 56 0 55 1;
495 #X connect 58 0 57 0;
#X connect 59 0 37 0;
#X connect 60 0 5 0;
#X connect 61 0 10 0;
#X connect 61 1 11 0;
500 #X connect 62 0 63 0;
```

```

#X connect 63 0 61 1;
#X restore 67 335 pd \ $0-worms;
#X obj 15 292 t a a;
#X obj 15 312 t a a;
505 #N canvas 0 50 319 140 \ $0-ground 0;
#X obj 17 12 inlet;
#X obj 71 12 inlet;
#X obj 19 46 without-lighting;
#X obj 18 80 ground;
510 #X connect 0 0 2 0;
#X connect 1 0 3 1;
#X connect 2 0 3 0;
#X restore 71 409 pd \ $0-ground;
#N canvas 389 50 299 402 \ $0-dolly 0;
515 #X obj 104 138 atan2;
#X obj 104 159 / 3.14159;
#X obj 38 222 / 2;
#X msg 103 111 100 \ $1;
#X obj 101 75 + 30;
520 #X obj 39 243 + 10;
#X obj 38 24 inlet;
#X obj 46 318 outlet;
#X obj 221 316 outlet;
#X msg 105 267 aperture \ $1;
525 #X msg 39 264 zero \ $1;
#X obj 104 185 * 180;
#X connect 0 0 1 0;
#X connect 1 0 11 0;
#X connect 2 0 5 0;
530 #X connect 3 0 0 0;
#X connect 4 0 3 0;
#X connect 4 0 8 0;
#X connect 5 0 10 0;
#X connect 6 0 4 0;
535 #X connect 6 0 2 0;
#X connect 9 0 7 0;
#X connect 10 0 7 0;
#X connect 11 0 9 0;
#X restore 55 95 pd \ $0-dolly;
540 #N canvas 3 44 482 453 \ $0-timeline 0;
#X obj 127 316 outlet;
#X obj 191 316 outlet;
#X obj 24 267 * 1000;
#X obj 21 94 f 0;
545 #X obj 23 135 / 1500;
#X obj 23 160 * 2;
#X obj 23 202 cos;
#X obj 20 62 spigot;
#X obj 53 42 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 1 1
550 ;
#X msg 78 53 0;
#X obj 23 181 * 3.14159;
#X obj 72 109 mod 3000;
#X obj 88 162 mod 3000;
555 #X obj 88 183 < 1500;
#X obj 71 88 + 1;
#X obj 161 66 * 60;

```

```
#X obj 161 87 * 25;
#X obj 161 113 / 2;
560 #X obj 160 134 / 2;
#X obj 196 184 s rec;
#X msg 207 53 \; pd dsp 1;
#X msg 53 23 1;
#X msg 229 132 1;
565 #X obj 199 89 * 1000;
#X obj 199 110 delay;
#X msg 199 131 0;
#X obj 196 158 spigot 0;
#X obj 415 12 inlet;
570 #X obj 130 45 t f b b b;
#X obj 114 4 inlet;
#X obj 14 3 inlet;
#X obj 9 317 outlet;
#X obj 117 23 t b f;
575 #X msg 105 82 1;
#X obj 75 272 swap 20;
#X obj 75 293 /;
#X obj 75 314 outlet;
#X obj 403 319 outlet;
580 #X obj 309 181 t f f;
#X obj 309 202 div 25;
#X obj 309 223 t f f;
#X obj 309 244 div 60;
#X obj 309 265 pack f f f;
585 #X obj 403 148 /;
#X obj 310 315 outlet;
#X obj 359 203 mod 25;
#X obj 355 244 mod 60;
#X obj 23 223 + 1;
590 #X obj 23 246 * 0.5;
#X obj 88 216 change -1;
#X obj 308 159 *;
#X obj 269 7 inlet;
#X obj 269 28 * 60;
595 #X obj 269 49 * 25;
#X connect 2 0 0 0;
#X connect 3 0 4 0;
#X connect 3 0 14 0;
#X connect 3 0 12 0;
600 #X connect 3 0 43 0;
#X connect 3 0 50 0;
#X connect 4 0 5 0;
#X connect 5 0 10 0;
#X connect 6 0 47 0;
605 #X connect 7 0 3 0;
#X connect 8 0 7 1;
#X connect 9 0 3 1;
#X connect 10 0 6 0;
#X connect 11 0 3 1;
610 #X connect 12 0 13 0;
#X connect 13 0 49 0;
#X connect 14 0 11 0;
#X connect 15 0 16 0;
#X connect 15 0 23 0;
```

```

615  #X connect 16 0 17 0;
      #X connect 16 0 11 1;
      #X connect 16 0 12 1;
      #X connect 16 0 43 1;
      #X connect 17 0 13 1;
620  #X connect 17 0 4 1;
      #X connect 17 0 18 0;
      #X connect 21 0 8 0;
      #X connect 22 0 26 0;
      #X connect 23 0 24 0;
625  #X connect 24 0 25 0;
      #X connect 25 0 26 0;
      #X connect 26 0 19 0;
      #X connect 27 0 26 1;
      #X connect 28 0 15 0;
630  #X connect 28 1 22 0;
      #X connect 28 2 20 0;
      #X connect 28 3 21 0;
      #X connect 28 3 9 0;
      #X connect 29 0 32 0;
635  #X connect 30 0 7 0;
      #X connect 32 0 33 0;
      #X connect 32 1 28 0;
      #X connect 32 1 34 0;
      #X connect 33 0 31 0;
640  #X connect 34 0 35 0;
      #X connect 34 1 35 1;
      #X connect 35 0 36 0;
      #X connect 35 0 50 1;
      #X connect 38 0 39 0;
645  #X connect 38 1 45 0;
      #X connect 39 0 40 0;
      #X connect 40 0 41 0;
      #X connect 40 1 46 0;
      #X connect 41 0 42 0;
650  #X connect 42 0 44 0;
      #X connect 43 0 37 0;
      #X connect 45 0 42 2;
      #X connect 46 0 42 1;
      #X connect 47 0 48 0;
655  #X connect 48 0 2 0;
      #X connect 49 0 1 0;
      #X connect 50 0 38 0;
      #X connect 51 0 52 0;
      #X connect 52 0 53 0;
660  #X connect 53 0 3 1;
      #X restore 41 49 pd \ $0-timeline;
      #X msg 47 7 2;
      #X msg 80 6 5;
      #X msg 113 5 10;
665  #X msg 145 7 20;
      #X obj 153 53 tgl 15 0 empty empty record 17 7 0 10 -262144 -1 -1 0
      1;
      #N canvas 363 44 632 360 \ $0-gemwin 0;
      #X obj 83 39 gemhead;
670  #X msg 322 108 0 \, destroy;
      #X obj 151 177 world.light;

```

```
#X obj 172 22 loadbang;
#X obj 259 140 gemwin 25;
#X obj 83 60 t a a;
675 #X obj 10 156 pix_write;
#X obj 34 83 r rec;
#X msg 34 104 auto \$1;
#X obj 342 13 inlet;
#X obj 361 44 sel 0 1;
680 #X obj 98 79 t a a b b;
#X obj 93 204 outlet;
#X obj 293 273 outlet;
#X obj 414 271 outlet;
#X msg 10 49 file o 0;
685 #X obj 11 19 loadbang;
#X obj 134 124 separator;
#X obj 140 146 rotateXYZ 60 120 0;
#X obj 156 274 world_light;
#X obj 139 221 separator;
690 #X obj 145 244 rotateXYZ 60 -120 0;
#X msg 168 67 far 10000 \, near 3;
#X obj 115 101 t a a a;
#X obj 431 214 world_light;
#X obj 414 161 separator;
695 #X obj 420 183 rotateXYZ 60 0 0;
#X obj 266 198 f 1;
#X obj 267 240 until;
#X obj 266 218 max 1;
#X obj 448 13 inlet;
700 #X msg 268 81 border 0 \, frame 25 \, offset 86 0 \, create \, 1 \,
lighting 1;
#X msg 242 49 dimen 1280 768;
#X connect 0 0 5 0;
#X connect 1 0 4 0;
705 #X connect 3 0 22 0;
#X connect 3 0 32 0;
#X connect 5 0 6 0;
#X connect 5 1 11 0;
#X connect 7 0 8 0;
710 #X connect 8 0 6 0;
#X connect 9 0 10 0;
#X connect 10 0 1 0;
#X connect 10 1 31 0;
#X connect 11 0 12 0;
715 #X connect 11 1 23 0;
#X connect 11 2 27 0;
#X connect 11 3 14 0;
#X connect 15 0 6 0;
#X connect 16 0 15 0;
720 #X connect 17 0 18 0;
#X connect 18 0 2 0;
#X connect 20 0 21 0;
#X connect 21 0 19 0;
#X connect 22 0 12 0;
725 #X connect 23 0 20 0;
#X connect 23 1 17 0;
#X connect 23 2 25 0;
#X connect 25 0 26 0;
```

```
#X connect 26 0 24 0;
730 #X connect 27 0 29 0;
#X connect 28 0 13 0;
#X connect 29 0 28 0;
#X connect 30 0 27 1;
#X connect 31 0 4 0;
735 #X connect 32 0 4 0;
#X connect 32 0 12 0;
#X restore 2 73 pd \${0-gemwin};
#X obj 9 50 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0 1
;
740 #X obj 15 192 ambientRGB 0.1 0.1 0.1;
#X obj 15 543 GEMglGetError;
#X text 34 563 FIXME workaround for console spam;
#X floatatom 179 115 5 1 10 1 overdrive - -, f 5;
#X text 180 7 duration (minutes);
745 #X msg 228 61 1;
#N canvas 293 384 450 610 \${0-ducklings 0;
#X obj 22 22 inlet;
#X obj 22 43 separator;
#X obj 283 21 inlet;
750 #X obj 21 101 t a a;
#X obj 46 120 repeat-gemlist 5;
#X obj 46 141 separator;
#X obj 152 104 sel 0;
#X obj 153 126 f;
755 #X obj 154 168 * 3.14159;
#X obj 154 189 cos;
#X obj 154 210 - 1;
#X obj 141 275 +;
#X obj 141 299 * 3.14159;
760 #X obj 78 241 +;
#X obj 64 265 * 3.14159;
#X obj 18 160 t a b;
#X obj 16 185 separator;
#X msg 83 185 5;
765 #X obj 141 399 * 100;
#X obj 61 349 * 100;
#X obj 141 254 * 0.01;
#X obj 80 209 * 0.01;
#X obj 154 231 * 0.25;
770 #X obj 14 530 model;
#X msg 113 501 rescale 1 \, open teapot.obj;
#X obj 115 480 loadbang;
#X obj 5 402 scale 4;
#X obj 53 454 scale 2;
775 #X obj 153 147 * 50;
#X obj 266 265 loadbang;
#X obj 61 327 sin;
#X obj 141 377 sin;
#X obj 149 19 inlet;
780 #X obj 149 40 * -1;
#X obj 251 69 * 2;
#X obj 142 356 * 6;
#X obj 62 306 * 6;
#X obj 5 379 translateXYZ 0 -4 0;
785 #X obj 53 433 translateXYZ 0 -5 0;
```

```
#X obj 20 84 translateXYZ 0 0 -50;
#X msg 264 285 70;
#X obj 251 93 - 0.25;
#X obj 330 48 * 2;
790 #X obj 330 69 wrap;
#X obj 22 64 spigot;
#X obj 330 90 - 0.5;
#X obj 382 159 < 0.15;
#X obj 330 111 t f f;
795 #X obj 333 158 &&;
#X obj 333 137 > -0.19;
#X obj 13 481 scaleXYZ 1 1 1;
#X obj 356 277 < 0;
#X obj 356 298 sel 1 0;
800 #X msg 356 319 -1;
#X msg 388 318 1;
#X connect 0 0 1 0;
#X connect 1 0 44 0;
#X connect 2 0 34 0;
805 #X connect 2 0 42 0;
#X connect 3 0 15 0;
#X connect 3 1 4 0;
#X connect 4 0 5 0;
#X connect 4 1 6 0;
810 #X connect 5 0 38 0;
#X connect 6 0 7 0;
#X connect 6 1 20 0;
#X connect 7 0 28 0;
#X connect 8 0 9 0;
815 #X connect 9 0 10 0;
#X connect 10 0 22 0;
#X connect 11 0 12 0;
#X connect 12 0 35 0;
#X connect 13 0 14 0;
820 #X connect 14 0 36 0;
#X connect 15 0 16 0;
#X connect 15 1 17 0;
#X connect 16 0 37 0;
#X connect 17 0 21 0;
825 #X connect 18 0 38 1;
#X connect 19 0 37 1;
#X connect 20 0 11 0;
#X connect 21 0 13 0;
#X connect 22 0 20 0;
830 #X connect 24 0 23 0;
#X connect 25 0 24 0;
#X connect 26 0 50 0;
#X connect 27 0 50 0;
#X connect 28 0 8 0;
835 #X connect 29 0 40 0;
#X connect 30 0 19 0;
#X connect 31 0 18 0;
#X connect 32 0 33 0;
#X connect 34 0 41 0;
840 #X connect 35 0 31 0;
#X connect 36 0 30 0;
#X connect 37 0 26 0;
```

```

#X connect 38 0 27 0;
#X connect 39 0 3 0;
845 #X connect 40 0 18 1;
#X connect 40 0 19 1;
#X connect 41 0 7 1;
#X connect 41 0 11 1;
#X connect 41 0 13 1;
850 #X connect 42 0 43 0;
#X connect 43 0 45 0;
#X connect 44 0 39 0;
#X connect 45 0 47 0;
#X connect 46 0 48 1;
855 #X connect 47 0 49 0;
#X connect 47 1 46 0;
#X connect 47 1 51 0;
#X connect 48 0 44 1;
#X connect 49 0 48 0;
860 #X connect 50 0 23 0;
#X connect 51 0 52 0;
#X connect 52 0 53 0;
#X connect 52 1 54 0;
#X connect 53 0 50 1;
865 #X connect 54 0 50 1;
#X restore 184 280 pd \${0}-ducklings;
#X obj 15 251 t a a;
#N canvas 0 50 450 300 \${0}-starfield 0;
#X obj 21 70 starfield;
870 #X obj 21 14 inlet;
#X connect 1 0 0 0;
#X restore 59 223 pd \${0}-starfield;
#X obj 15 213 t a a;
#X obj 193 420 *~ 0.125;
875 #X obj 252 421 *~ 0.125;
#N canvas 0 50 450 300 \${0}-way-markers 0;
#X obj 21 18 inlet;
#X obj 21 69 repeat-gemlist 20;
#X obj 118 89 * -50;
880 #X obj 21 90 separator;
#X obj 20 182 text3d;
#X obj 206 81 loadbang;
#X msg 206 102 font /usr/share/fonts/truetype/liberation/LiberationMono-Bold.ttf
;
885 #X obj 21 131 scaleXYZ -1 1 1;
#X obj 21 110 translateXYZ 0 -6 0;
#X msg 118 158 text \${1} \${2};
#X obj 119 130 pack f f;
#X obj 21 39 repeat-gemlist 5;
890 #X obj 148 49 - 2;
#X obj 148 70 * 20;
#X connect 0 0 11 0;
#X connect 1 0 3 0;
#X connect 1 1 2 0;
895 #X connect 2 0 8 3;
#X connect 2 0 10 0;
#X connect 3 0 8 0;
#X connect 5 0 6 0;
#X connect 6 0 4 0;

```



```

900  #X connect 7 0 4 0;
    #X connect 8 0 7 0;
    #X connect 9 0 4 0;
    #X connect 10 0 9 0;
    #X connect 11 0 1 0;
905  #X connect 11 1 12 0;
    #X connect 12 0 13 0;
    #X connect 13 0 8 1;
    #X connect 13 0 10 1;
    #X restore 195 516 pd \ $0-way-markers;
910  #X obj 79 482 spigot;
    #X obj 136 487 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
    1;
    #N canvas 607 108 450 300 \ $0-clock 0;
    #X obj 27 21 inlet;
915  #X obj 27 42 separator;
    #X obj 27 131 text3d;
    #X obj 171 22 inlet;
    #X obj 84 98 loadbang;
    #X msg 82 118 font /usr/share/fonts/truetype/liberation/LiberationMono-Bold.ttf
920  ;
    #X obj 28 84 scaleXYZ -1 1 1;
    #X obj 27 63 translateXYZ 20 6 0;
    #X obj 28 105 scale 4;
    #X msg 171 43 text \ $1 \ $2;
925  #X connect 0 0 1 0;
    #X connect 1 0 7 0;
    #X connect 3 0 9 0;
    #X connect 4 0 5 0;
    #X connect 5 0 2 0;
930  #X connect 6 0 8 0;
    #X connect 7 0 6 0;
    #X connect 8 0 2 0;
    #X connect 9 0 2 0;
    #X restore 108 519 pd \ $0-clock;
935  #X obj 163 85 spigot 1;
    #X obj 226 88 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
    1;
    #X msg 13 8 1;
    #X obj 187 489 spigot;
940  #X obj 244 494 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
    1;
    #X obj 17 458 t a a a;
    #X floatatom 181 27 5 0 0 0 - - -, f 5;
    #X connect 0 0 1 0;
945  #X connect 1 0 3 0;
    #X connect 2 0 47 0;
    #X connect 2 1 6 0;
    #X connect 3 0 26 0;
    #X connect 7 0 2 0;
950  #X connect 7 1 16 0;
    #X connect 8 0 11 0;
    #X connect 8 1 9 0;
    #X connect 9 0 36 0;
    #X connect 9 1 37 0;
955  #X connect 11 0 7 0;
    #X connect 11 1 10 0;

```

```

#X connect 12 0 36 0;
#X connect 12 1 37 0;
#X connect 13 0 36 0;
960 #X connect 13 1 37 0;
#X connect 14 0 15 0;
#X connect 14 1 12 0;
#X connect 15 0 8 0;
#X connect 15 1 13 0;
965 #X connect 17 0 0 0;
#X connect 17 1 1 3;
#X connect 17 1 32 1;
#X connect 18 0 25 0;
#X connect 18 1 42 0;
970 #X connect 18 2 9 2;
#X connect 18 2 12 1;
#X connect 18 2 13 1;
#X connect 18 2 6 1;
#X connect 18 2 17 0;
975 #X connect 18 3 16 1;
#X connect 18 4 41 1;
#X connect 18 5 32 2;
#X connect 18 5 9 3;
#X connect 18 5 13 2;
980 #X connect 18 5 12 2;
#X connect 19 0 18 1;
#X connect 20 0 18 1;
#X connect 21 0 18 1;
#X connect 22 0 18 1;
985 #X connect 23 0 18 3;
#X connect 24 0 0 0;
#X connect 24 1 9 1;
#X connect 24 2 18 0;
#X connect 25 0 24 0;
990 #X connect 26 0 35 0;
#X connect 29 0 24 1;
#X connect 31 0 29 0;
#X connect 33 0 14 0;
#X connect 33 1 32 0;
995 #X connect 35 0 33 0;
#X connect 35 1 34 0;
#X connect 36 0 4 0;
#X connect 37 0 5 0;
#X connect 39 0 41 0;
1000 #X connect 40 0 39 1;
#X connect 42 0 29 0;
#X connect 43 0 42 1;
#X connect 44 0 18 1;
#X connect 45 0 38 0;
1005 #X connect 46 0 45 1;
#X connect 47 0 27 0;
#X connect 47 1 39 0;
#X connect 47 2 45 0;
#X connect 48 0 18 2;

```

10 prototype/anaglyph.pd

```
#N canvas 3 44 610 698 10;
```

```

#X obj 22 22 inlet;
#X obj 22 84 unpack f f;
#X obj 22 105 /;
5 #X obj 119 85 unpack f;
#X obj 208 87 unpack f;
#X obj 286 91 unpack f;
#X obj 355 91 unpack f;
#X obj 415 91 unpack f;
10 #X obj 22 43 route dimen aperture eyesep near zero far;
#X obj 119 108 v \${0}-aperture;
#X obj 22 126 v \${0}-aspect;
#X obj 224 416 GEMglLoadIdentity;
#X obj 225 468 GEMglFrustum;
15 #X obj 225 560 GEMglLoadIdentity;
#X obj 14 417 GEMglLoadIdentity;
#X obj 14 458 GEMglFrustum;
#X obj 14 500 GEMglLoadIdentity;
#X obj 122 676 outlet;
20 #X obj 359 449 r \${0}-near;
#X obj 424 449 r \${0}-far;
#X obj 415 112 s \${0}-far;
#X obj 286 112 s \${0}-near;
#X obj 355 112 s \${0}-zero;
25 #X obj 103 458 r \${0}-near;
#X obj 168 458 r \${0}-far;
#X obj 14 542 GEMglColorMask 1 0 0 0;
#X obj 440 159 / 2;
#X obj 439 227 tan;
30 #X obj 440 279 *;
#X obj 470 226 r \${0}-near;
#X obj 441 304 * -1;
#X obj 216 177 v \${0}-aspect;
#X obj 440 138 v \${0}-aperture;
35 #X obj 207 110 v \${0}-eyesep;
#X obj 216 284 +;
#X obj 255 285 -;
#X obj 328 200 r \${0}-near;
#X obj 328 221 r \${0}-zero;
40 #X obj 220 319 * -1;
#X obj 261 309 * -1;
#X obj 299 240 /;
#X obj 296 196 / 2;
#X obj 133 14 loadbang;
45 #X obj 225 605 GEMgluLookAt 0 0 0 0 0 1 0 1 0;
#X obj 440 180 / 180;
#X obj 440 201 * 3.14159;
#X obj 296 136 v \${0}-eyesep;
#X obj 14 396 GEMglMatrixMode;
50 #X obj 393 356 loadbang;
#X obj 347 398 GLdefine GL_PROJECTION;
#X obj 224 395 GEMglMatrixMode;
#X obj 225 505 GEMglMatrixMode;
#X obj 14 479 GEMglMatrixMode;
55 #X obj 351 506 GLdefine GL_MODELVIEW;
#X obj 74 177 separator;
#X obj 98 203 separator;
#X obj 15 152 t a a;

```

```
#X obj 117 574 * -1;
60 #X obj 370 554 * -1;
#X obj 14 521 GEMgluLookAt 0 0 0 0 0 1 0 1 0;
#X obj 26 327 GEMglClear;
#X obj 400 380 GLdefine GL_DEPTH_BUFFER_BIT;
#X obj 74 156 t a a a;
65 #X obj 199 142 t a b b b;
#X obj 226 647 GEMglColorMask 0 1 0 0;
#X obj 14 235 GEMglColorMask 1 1 0 1;
#X obj 217 258 *;
#X obj 299 219 *;
70 #X msg 239 10 dimen 1280 768 \, aperture 30 \, eyesep 1 \, near 10
\, zero 100 \, far 10000;
#X obj 439 255 * 0.8;
#X connect 0 0 8 0;
#X connect 1 0 2 0;
75 #X connect 1 1 2 1;
#X connect 2 0 10 0;
#X connect 3 0 9 0;
#X connect 4 0 33 0;
#X connect 5 0 21 0;
80 #X connect 6 0 22 0;
#X connect 7 0 20 0;
#X connect 8 0 1 0;
#X connect 8 1 3 0;
#X connect 8 2 4 0;
85 #X connect 8 3 5 0;
#X connect 8 4 6 0;
#X connect 8 5 7 0;
#X connect 8 6 56 0;
#X connect 11 0 12 0;
90 #X connect 12 0 51 0;
#X connect 13 0 43 0;
#X connect 14 0 15 0;
#X connect 15 0 52 0;
#X connect 16 0 59 0;
95 #X connect 18 0 12 5;
#X connect 19 0 12 6;
#X connect 23 0 15 5;
#X connect 24 0 15 6;
#X connect 25 0 17 0;
100 #X connect 26 0 44 0;
#X connect 27 0 69 0;
#X connect 28 0 30 0;
#X connect 28 0 15 4;
#X connect 28 0 12 4;
105 #X connect 28 0 66 1;
#X connect 29 0 28 1;
#X connect 30 0 12 3;
#X connect 30 0 15 3;
#X connect 31 0 66 0;
110 #X connect 32 0 26 0;
#X connect 34 0 38 0;
#X connect 34 0 15 2;
#X connect 35 0 39 0;
#X connect 35 0 12 2;
115 #X connect 36 0 67 1;
```

```

#X connect 37 0 40 1;
#X connect 38 0 12 1;
#X connect 39 0 15 1;
#X connect 40 0 34 1;
120 #X connect 40 0 35 1;
#X connect 41 0 67 0;
#X connect 41 0 58 0;
#X connect 42 0 68 0;
#X connect 43 0 64 0;
125 #X connect 44 0 45 0;
#X connect 45 0 27 0;
#X connect 46 0 41 0;
#X connect 47 0 14 0;
#X connect 48 0 49 0;
130 #X connect 48 0 53 0;
#X connect 48 0 61 0;
#X connect 49 0 47 1;
#X connect 49 0 50 1;
#X connect 50 0 11 0;
135 #X connect 51 0 13 0;
#X connect 52 0 16 0;
#X connect 53 0 51 1;
#X connect 53 0 52 1;
#X connect 54 0 60 0;
140 #X connect 55 0 50 0;
#X connect 56 0 65 0;
#X connect 56 1 62 0;
#X connect 57 0 59 1;
#X connect 57 0 59 4;
145 #X connect 58 0 43 1;
#X connect 58 0 43 4;
#X connect 58 0 57 0;
#X connect 59 0 25 0;
#X connect 60 0 47 0;
150 #X connect 61 0 60 1;
#X connect 62 0 54 0;
#X connect 62 1 55 0;
#X connect 62 2 63 0;
#X connect 63 1 31 0;
155 #X connect 63 2 46 0;
#X connect 63 3 32 0;
#X connect 64 0 17 0;
#X connect 66 0 34 0;
#X connect 66 0 35 0;
160 #X connect 67 0 40 0;
#X connect 68 0 8 0;
#X connect 69 0 28 0;

```

11 prototype/ball-sound.pd

```

#N canvas 525 265 450 611 10;
#X obj 31 25 inlet;
#X obj 78 390 outlet;
#X obj 107 184 t a a b;
5 #X obj 226 205 vec3-add;
#X obj 373 36 inlet;
#X obj 226 226 vec3-length;

```

```

#X obj 259 403 vline~;
#X obj 232 336 expr 900 / ($f1 * $f1 + 0.001);
10 #X obj 223 470 *~;
#X obj 145 470 *~;
#X obj 103 208 vec3-add;
#X obj 103 229 vec3-length;
#X obj 136 406 vline~;
15 #X obj 98 317 expr 900 / ($f1 * $f1 + 0.001);
#X obj 31 46 route \$2;
#X msg 259 382 \$1 \, 0 10;
#X msg 136 385 \$1 \, 0 10;
#X obj 188 431 noise~;
20 #X obj 145 491 throw~ \$1-l;
#X obj 223 491 throw~ \$1-r;
#X obj 351 494 outlet;
#X obj 31 88 pack f f 0;
#X obj 31 147 spigot;
25 #X obj 105 112 t f f b;
#X obj 133 138 f;
#X obj 112 162 !=;
#X obj 31 67 unpack f f f f;
#X obj 108 251 /;
30 #X obj 229 252 /;
#X obj 104 88 >= 0;
#X obj 98 338 clip 0 3;
#X obj 236 357 clip 0 3;
#X msg 313 176 20 0 \$1;
35 #X msg 187 183 -20 0 \$1;
#X obj 134 87 +;
#X obj 229 273 + 20;
#X obj 108 272 + 20;
#X connect 0 0 14 0;
40 #X connect 2 0 10 0;
#X connect 2 1 3 0;
#X connect 3 0 5 0;
#X connect 4 0 32 0;
#X connect 4 0 33 0;
45 #X connect 4 0 20 0;
#X connect 5 0 28 0;
#X connect 6 0 8 1;
#X connect 7 0 31 0;
#X connect 8 0 19 0;
50 #X connect 9 0 18 0;
#X connect 10 0 11 0;
#X connect 11 0 27 0;
#X connect 12 0 9 1;
#X connect 13 0 30 0;
55 #X connect 14 0 26 0;
#X connect 14 1 1 0;
#X connect 15 0 6 0;
#X connect 16 0 12 0;
#X connect 17 0 8 0;
60 #X connect 17 0 9 0;
#X connect 21 0 22 0;
#X connect 22 0 2 0;
#X connect 23 0 25 0;
#X connect 23 1 24 1;

```

```

65  #X connect 23 2 24 0;
    #X connect 24 0 25 1;
    #X connect 25 0 22 1;
    #X connect 26 0 21 0;
    #X connect 26 1 21 1;
70  #X connect 26 2 29 0;
    #X connect 26 3 34 0;
    #X connect 27 0 36 0;
    #X connect 28 0 35 0;
    #X connect 29 0 23 0;
75  #X connect 30 0 16 0;
    #X connect 31 0 15 0;
    #X connect 32 0 3 1;
    #X connect 33 0 10 1;
    #X connect 34 0 27 1;
80  #X connect 34 0 28 1;
    #X connect 35 0 7 0;
    #X connect 36 0 13 0;

```

12 prototype/ball-sounds.pd

```

#N canvas 0 0 450 300 10;
#X obj 18 22 inlet;
#X obj 129 25 inlet;
#X obj 208 108 ~ 0.25;
5  #X obj 209 129 cos~;
    #X obj 270 110 ~ 0.25;
    #X obj 270 131 cos~;
    #X obj 209 68 vcf~ 200;
    #X obj 271 67 vcf~ 200;
10  #X obj 207 33 catch~ \ $0-l;
    #X obj 288 39 catch~ \ $0-r;
    #X obj 209 150 outlet~;
    #X obj 270 152 outlet~;
    #X obj 18 63 ball-sound \ $0 0;
15  #X obj 262 11 inlet;
    #X obj 18 123 ball-sound \ $0 3;
    #X obj 18 83 ball-sound \ $0 1;
    #X obj 18 103 ball-sound \ $0 2;
    #X obj 209 89 *~ 1000;
20  #X obj 272 88 *~ 1000;
    #X connect 0 0 12 0;
    #X connect 1 0 12 1;
    #X connect 2 0 3 0;
    #X connect 3 0 10 0;
25  #X connect 4 0 5 0;
    #X connect 5 0 11 0;
    #X connect 6 0 17 0;
    #X connect 7 0 18 0;
    #X connect 8 0 6 0;
30  #X connect 9 0 7 0;
    #X connect 12 0 15 0;
    #X connect 12 1 15 1;
    #X connect 13 0 7 1;
    #X connect 13 0 6 1;
35  #X connect 15 0 16 0;
    #X connect 15 1 16 1;

```

```

#X connect 16 0 14 0;
#X connect 16 1 14 1;
#X connect 17 0 2 0;
40 #X connect 18 0 4 0;

```

13 prototype/encode.sh

```

#!/bin/bash
stem="zoon-$(date --iso)"
avconv -i ../sketches/thunder/o.wav -i o%05d.tif -vf vflip -s 384x288 -vb 1M -ab 192k -acodec libvorbis "${stem}.ogv"
avconv -i ../sketches/thunder/o.wav -i o%05d.tif -vf vflip -target dvd -aspect 4:3 "${stem}.mpeg"

```

14 prototype/geo-pipe.pd

```

#N canvas 917 544 450 387 10;
#X obj 38 28 inlet;
#X obj 113 19 inlet;
#X obj 211 19 inlet;
5 #X obj 321 22 inlet;
#X obj 113 44 unpack f f f;
#X obj 211 44 unpack f f f;
#X obj 173 102 -;
#X obj 203 102 -;
10 #X obj 43 62 separator;
#X obj 172 150 t f f;
#X obj 173 174 *;
#X obj 44 98 translate 1 0 0 0;
#X obj 46 223 rotate;
15 #X obj 212 150 t f f;
#X obj 213 174 *;
#X obj 262 150 t f f;
#X obj 263 174 *;
#X obj 218 199 +;
20 #X obj 174 202 +;
#X obj 174 225 sqrt;
#X obj 174 248 swap;
#X obj 175 272 /;
#X obj 174 299 expr 180.0*acos($f1)/3.1415926;
25 #X obj 174 329 *;
#X obj 211 317 expr if($f1<0 \, -1 \, 1);
#X obj 93 196 pack f f;
#X msg 93 221 \"$2 \"$1 0;
#X obj 138 174 * -1;
30 #X obj 139 154 *;
#X obj 93 156 *;
#X obj 231 115 +;
#X obj 248 66 -;
#X obj 275 109 == 0;
35 #X obj 277 129 * 1e-05;
#X obj 250 87 t f f;
#X obj 293 266 * 0.5;
#X obj 31 271 scaleXYZ -1 1 1;
#X obj 81 320 separator;
40 #X obj 35 249 separator;

```



```
#X obj 82 341 outlet;
#X obj 17 335 outlet;
#X obj 293 243 / 1;
#X obj 32 293 translateXYZ 0 0 1;
45 #X connect 0 0 8 0;
#X connect 1 0 4 0;
#X connect 2 0 5 0;
#X connect 2 0 11 2;
#X connect 3 0 41 1;
50 #X connect 3 0 42 3;
#X connect 4 0 6 0;
#X connect 4 1 7 0;
#X connect 4 2 31 0;
#X connect 5 0 6 1;
55 #X connect 5 1 7 1;
#X connect 5 2 31 1;
#X connect 6 0 9 0;
#X connect 6 0 29 0;
#X connect 7 0 13 0;
60 #X connect 7 0 28 0;
#X connect 8 0 11 0;
#X connect 9 0 10 0;
#X connect 9 1 10 1;
#X connect 10 0 18 0;
65 #X connect 11 0 12 0;
#X connect 12 0 38 0;
#X connect 12 0 37 0;
#X connect 13 0 14 0;
#X connect 13 1 14 1;
70 #X connect 14 0 17 0;
#X connect 15 0 16 0;
#X connect 15 1 16 1;
#X connect 16 0 17 1;
#X connect 17 0 18 1;
75 #X connect 18 0 19 0;
#X connect 19 0 20 0;
#X connect 19 0 41 0;
#X connect 20 0 21 0;
#X connect 20 1 21 1;
80 #X connect 21 0 22 0;
#X connect 22 0 23 0;
#X connect 23 0 12 1;
#X connect 24 0 23 1;
#X connect 25 0 26 0;
85 #X connect 26 0 12 2;
#X connect 27 0 25 1;
#X connect 28 0 27 0;
#X connect 29 0 25 0;
#X connect 30 0 15 0;
90 #X connect 30 0 20 1;
#X connect 30 0 24 0;
#X connect 30 0 28 1;
#X connect 30 0 29 1;
#X connect 31 0 34 0;
95 #X connect 32 0 33 0;
#X connect 33 0 30 1;
#X connect 34 0 30 0;
```

```

#X connect 34 1 32 0;
#X connect 35 0 36 3;
100 #X connect 36 0 42 0;
#X connect 37 0 39 0;
#X connect 38 0 36 0;
#X connect 41 0 35 0;
#X connect 42 0 40 0;

```

15 prototype/ground.pd

```

#N canvas 665 327 1166 620 10;
#X obj 34 18 inlet;
#X obj 227 19 inlet;
#X obj 34 39 separator;
5 #X obj 34 83 translateXYZ 0 -6 0;
#X obj 34 104 rotateXYZ 90 0 0;
#X obj 421 200 tabread \${0}-t;
#X obj 575 271 tabread \${0}-r;
#X obj 502 162 t f f f f;
10 #X obj 381 287 mod 60;
#X obj 299 351 tabread \${0}-px;
#X obj 299 372 *;
#X obj 397 354 tabread \${0}-py;
#X obj 397 375 *;
15 #X obj 299 393 +;
#X obj 397 396 +;
#X obj 484 313 tabread \${0}-x;
#X obj 565 317 tabread \${0}-y;
#X obj 432 287 mod 60;
20 #X obj 495 355 tabread \${0}-px;
#X obj 495 376 *;
#X obj 590 355 tabread \${0}-py;
#X obj 590 376 *;
#X obj 495 397 +;
25 #X obj 590 397 +;
#X obj 561 10 loadbang;
#X obj 690 109 f 0;
#X obj 772 105 + 1;
#X obj 618 110 t b f;
30 #X obj 618 134 random 1e+06;
#X obj 618 155 / 1e+06;
#X obj 618 176 * 2;
#X obj 618 197 - 1;
#X obj 618 231 tabwrite \${0}-x;
35 #X obj 707 138 t b f;
#X obj 707 162 random 1e+06;
#X obj 707 183 / 1e+06;
#X obj 707 204 * 2;
#X obj 707 225 - 1;
40 #X obj 707 246 tabwrite \${0}-y;
#X obj 793 146 t b f;
#X obj 793 170 random 60;
#X obj 795 243 tabwrite \${0}-t;
#X obj 899 104 t b f;
45 #X obj 899 125 random 1e+06;
#X obj 899 146 / 1e+06;
#X obj 899 210 t f f;

```

```

#X obj 899 231 *;
#X obj 900 276 tabwrite \ $0-r;
50 #X obj 1004 29 loadbang;
#X msg 1004 50 60;
#X obj 1004 99 f 0;
#X obj 1054 100 + 1;
#X obj 1054 121 mod 60;
55 #X obj 1003 158 / 60;
#X obj 1003 179 * 2;
#X obj 1003 200 * 3.14159;
#X obj 1003 221 t f f;
#X obj 1003 242 cos;
60 #X obj 1033 244 sin;
#X obj 1004 120 t f f;
#X obj 1003 263 tabwrite \ $0-px;
#X obj 998 291 tabwrite \ $0-py;
#X obj 948 376 table \ $0-px 60;
65 #X obj 948 397 table \ $0-py 60;
#X obj 1004 71 until;
#X obj 681 93 until;
#X obj 561 33 t b b;
#X obj 381 54 until;
70 #X obj 436 74 + 1;
#X obj 381 96 t b f;
#X msg 381 117 60;
#X obj 381 138 until;
#X obj 436 151 + 1;
75 #X obj 436 172 mod 60;
#X obj 381 221 +;
#X obj 381 241 t f f;
#X obj 431 260 - 1;
#X obj 528 432 pack f f 0;
80 #X obj 338 431 pack f f 0;
#X obj 426 467 t a b;
#X obj 513 469 + 1;
#X obj 376 496 unpack f f f;
#X obj 376 517 tabwrite \ $0-vx;
85 #X obj 429 566 tabwrite \ $0-vz;
#X obj 400 538 tabwrite \ $0-vy;
#X obj 317 25 t b b;
#X obj 318 45 f \ $0;
#X msg 81 520 position \ $1-vx \ $1-vy \ $1-vz \, draw lines;
90 #X obj 381 75 f 0;
#X obj 381 158 f 0;
#X obj 474 468 f 0;
#X obj 432 308 t f f;
#X obj 381 308 t f f;
95 #X obj 899 191 + 0.5;
#X obj 899 167 * 0.5;
#X obj 34 125 scale 1000;
#X obj 34 187 rotateXYZ 0 0 90;
#X obj 34 146 t a a a a;
100 #X obj 900 256 * 0.005;
#X msg 681 72 2000;
#X obj 841 356 table \ $0-x 2000;
#X obj 841 377 table \ $0-y 2000;
#X obj 841 398 table \ $0-t 2000;

```

```

105  #X obj 841 419 table \${0-r 2000;
    #X obj 715 412 table \${0-vz 240000;
    #X obj 715 391 table \${0-vy 240000;
    #X obj 715 370 table \${0-vx 240000;
    #X obj 513 490 mod 240000;
110  #X obj 436 95 mod 2000;
    #X msg 382 33 2000;
    #X obj 772 126 mod 2000;
    #X obj 30 551 gemvertexbuffer 240000;
    #X obj 34 61 colorRGB 0.5 0.5 0.5;
115  #X connect 0 0 2 0;
    #X connect 2 0 112 0;
    #X connect 3 0 4 0;
    #X connect 4 0 95 0;
    #X connect 5 0 74 1;
120  #X connect 6 0 10 1;
    #X connect 6 0 12 1;
    #X connect 6 0 21 1;
    #X connect 6 0 19 1;
    #X connect 7 0 15 0;
125  #X connect 7 1 16 0;
    #X connect 7 2 5 0;
    #X connect 7 3 6 0;
    #X connect 8 0 92 0;
    #X connect 9 0 10 0;
130  #X connect 10 0 13 0;
    #X connect 11 0 12 0;
    #X connect 12 0 14 0;
    #X connect 13 0 78 0;
    #X connect 14 0 78 1;
135  #X connect 15 0 13 1;
    #X connect 15 0 22 1;
    #X connect 16 0 14 1;
    #X connect 16 0 23 1;
    #X connect 17 0 91 0;
140  #X connect 18 0 19 0;
    #X connect 19 0 22 0;
    #X connect 20 0 21 0;
    #X connect 21 0 23 0;
    #X connect 22 0 77 0;
145  #X connect 23 0 77 1;
    #X connect 24 0 66 0;
    #X connect 25 0 26 0;
    #X connect 25 0 27 0;
    #X connect 25 0 33 0;
150  #X connect 25 0 39 0;
    #X connect 25 0 42 0;
    #X connect 26 0 110 0;
    #X connect 27 0 28 0;
    #X connect 27 1 32 1;
155  #X connect 28 0 29 0;
    #X connect 29 0 30 0;
    #X connect 30 0 31 0;
    #X connect 31 0 32 0;
    #X connect 33 0 34 0;
160  #X connect 33 1 38 1;
    #X connect 34 0 35 0;

```

```

#X connect 35 0 36 0;
#X connect 36 0 37 0;
#X connect 37 0 38 0;
165 #X connect 39 0 40 0;
#X connect 39 1 41 1;
#X connect 40 0 41 0;
#X connect 42 0 43 0;
#X connect 42 1 47 1;
170 #X connect 43 0 44 0;
#X connect 44 0 94 0;
#X connect 45 0 46 0;
#X connect 45 1 46 1;
#X connect 46 0 98 0;
175 #X connect 48 0 49 0;
#X connect 49 0 64 0;
#X connect 50 0 51 0;
#X connect 50 0 59 0;
#X connect 51 0 52 0;
180 #X connect 52 0 50 1;
#X connect 53 0 54 0;
#X connect 54 0 55 0;
#X connect 55 0 56 0;
#X connect 56 0 57 0;
185 #X connect 56 1 58 0;
#X connect 57 0 60 0;
#X connect 58 0 61 0;
#X connect 59 0 53 0;
#X connect 59 1 60 1;
190 #X connect 59 1 61 1;
#X connect 64 0 50 0;
#X connect 65 0 25 0;
#X connect 66 0 85 0;
#X connect 66 1 49 0;
195 #X connect 66 1 99 0;
#X connect 67 0 88 0;
#X connect 68 0 108 0;
#X connect 69 0 70 0;
#X connect 69 1 7 0;
200 #X connect 70 0 71 0;
#X connect 71 0 89 0;
#X connect 72 0 73 0;
#X connect 73 0 89 1;
#X connect 74 0 75 0;
205 #X connect 75 0 8 0;
#X connect 75 1 76 0;
#X connect 76 0 17 0;
#X connect 77 0 79 0;
#X connect 78 0 79 0;
210 #X connect 79 0 81 0;
#X connect 79 1 90 0;
#X connect 80 0 107 0;
#X connect 81 0 82 0;
#X connect 81 1 84 0;
215 #X connect 81 2 83 0;
#X connect 85 0 86 0;
#X connect 85 1 109 0;
#X connect 86 0 87 0;

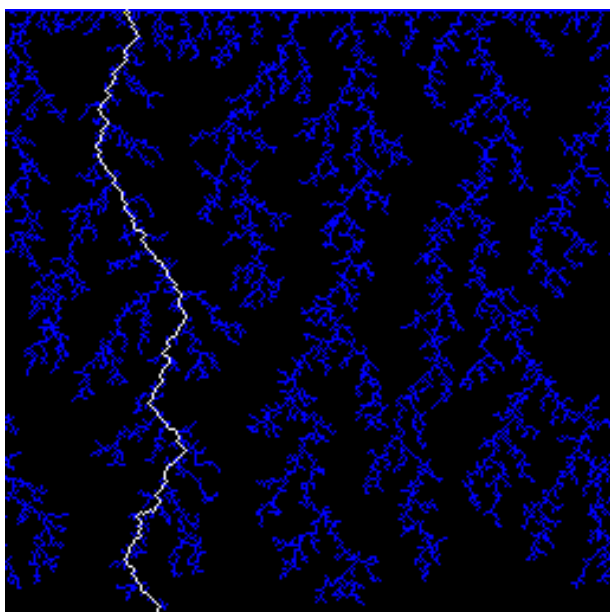
```

```

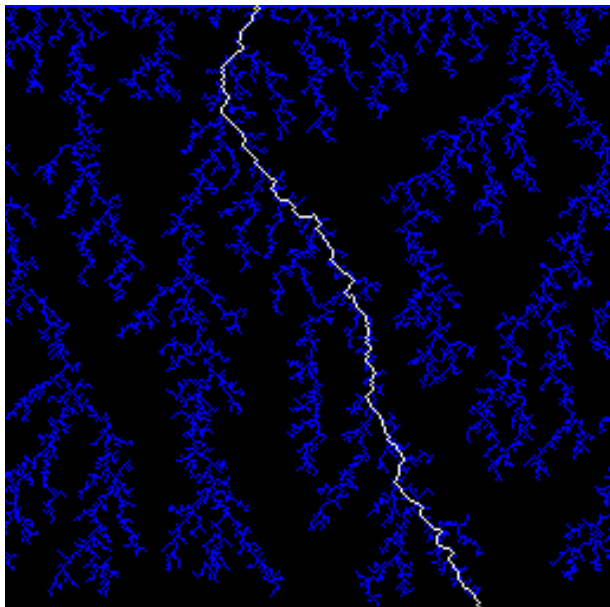
#X connect 87 0 111 0;
220 #X connect 88 0 68 0;
#X connect 88 0 69 0;
#X connect 89 0 72 0;
#X connect 89 0 74 0;
#X connect 90 0 80 0;
225 #X connect 90 0 82 1;
#X connect 90 0 84 1;
#X connect 90 0 83 1;
#X connect 91 0 18 0;
#X connect 91 1 20 0;
230 #X connect 92 0 9 0;
#X connect 92 1 11 0;
#X connect 93 0 45 0;
#X connect 94 0 93 0;
#X connect 95 0 97 0;
235 #X connect 96 0 111 0;
#X connect 97 0 96 0;
#X connect 97 1 96 0;
#X connect 97 2 96 0;
#X connect 97 3 96 0;
240 #X connect 98 0 47 0;
#X connect 99 0 65 0;
#X connect 107 0 90 1;
#X connect 108 0 88 1;
#X connect 109 0 67 0;
245 #X connect 110 0 25 1;
#X connect 112 0 3 0;

```

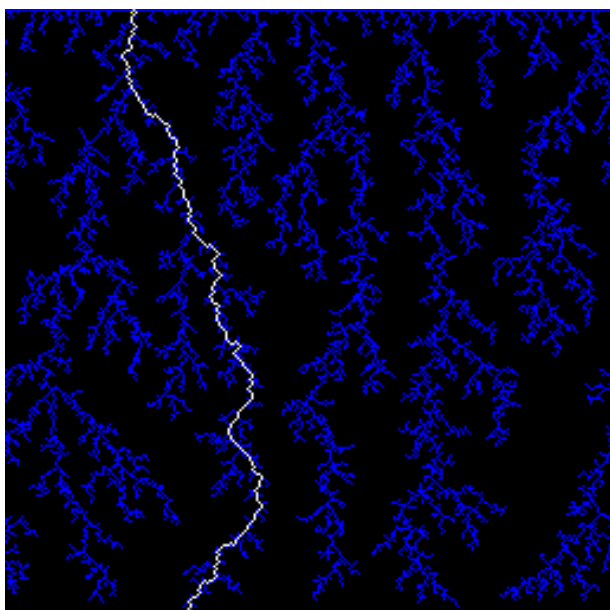
16 prototype/lightning-0.jpg



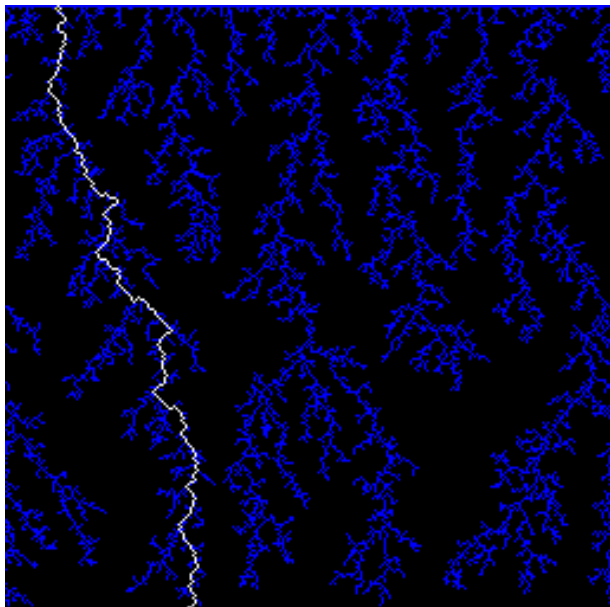
17 prototype/lightning-1.jpg



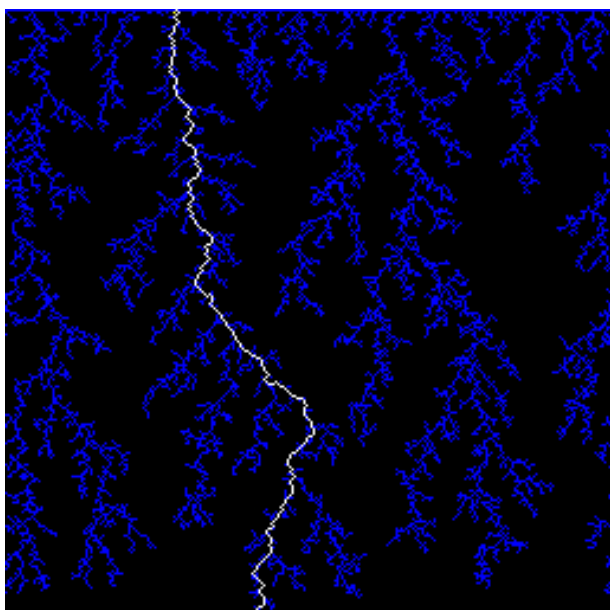
18 prototype/lightning-2.jpg



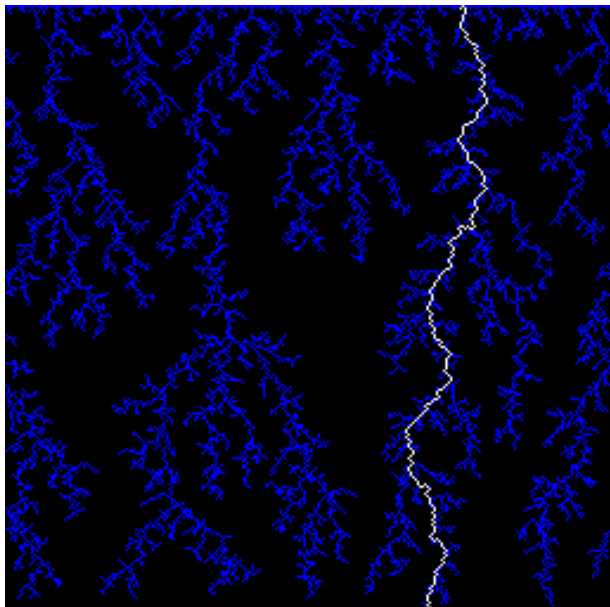
19 prototype/lightning-3.jpg



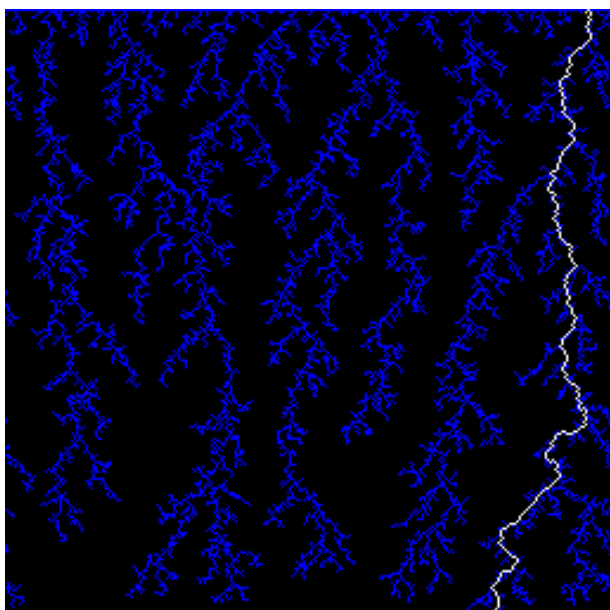
20 prototype/lightning-4.jpg



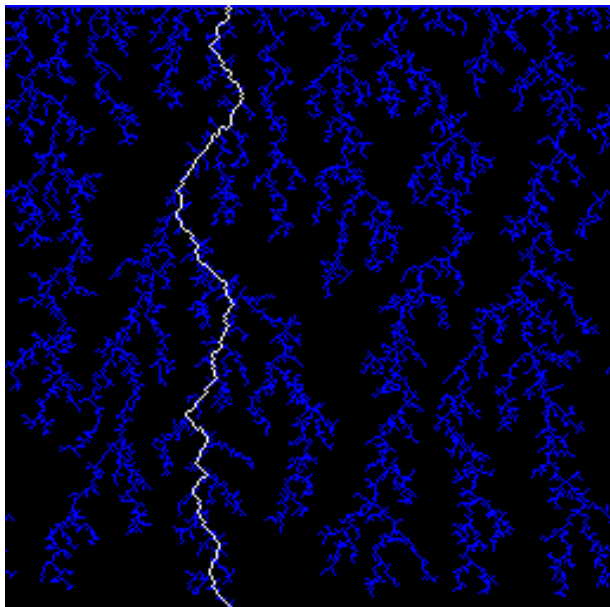
21 prototype/lightning-5.jpg



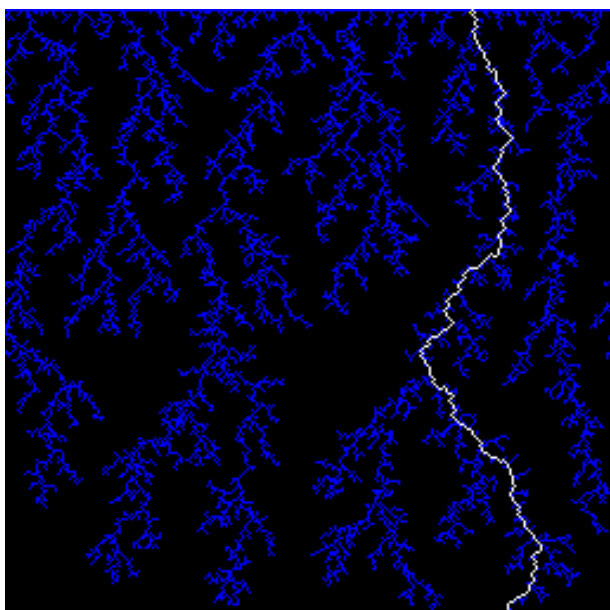
22 prototype/lightning-6.jpg



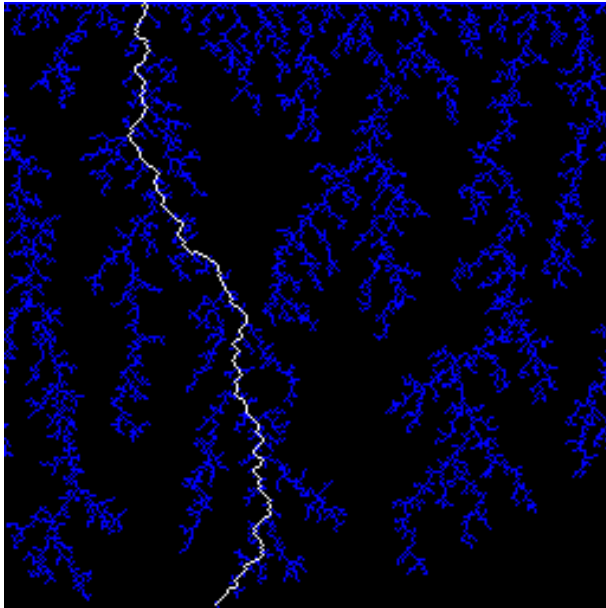
23 prototype/lightning-7.jpg



24 prototype/lightning-8.jpg



25 prototype/lightning-9.jpg



26 prototype/lightning.pd

```

#N canvas 501 137 300 489 10;
#X obj 97 26 inlet;
#X obj 32 24 inlet;
#X obj 30 56 separator;
5 #X obj 27 276 pix_texture;
#X obj 27 297 GEMglBlendFunc GL_SRC_ALPHA GL_ONE;
#X obj 28 367 square 50;
#X obj 27 231 translateXYZ 0 50 0;
#X obj 26 210 scaleXYZ 1 3 1;
10 #X obj 26 189 translateXYZ 0 0 200;
#X msg 103 83 1;
#X msg 149 102 0;
#X obj 149 81 delay 100;
#X obj 27 318 with_blend;
15 #X obj 27 339 without_lighting;
#X msg 222 104 0.3333;
#X obj 171 160 outlet;
#X obj 27 254 multi-texture lightning- .jpg;
#X obj 97 54 t f b b b b;
20 #X obj 225 160 random 10;
#X obj 29 142 colorRGB 0 0 0;
#X connect 0 0 17 0;
#X connect 1 0 2 0;
#X connect 2 0 19 0;
25 #X connect 3 0 4 0;
#X connect 4 0 12 0;
#X connect 6 0 16 0;
#X connect 7 0 6 0;
#X connect 8 0 7 0;
30 #X connect 9 0 19 3;

```

```

#X connect 9 0 19 2;
#X connect 9 0 19 1;
#X connect 10 0 15 0;
#X connect 10 0 19 3;
35 #X connect 10 0 19 2;
#X connect 10 0 19 1;
#X connect 11 0 10 0;
#X connect 12 0 13 0;
#X connect 13 0 5 0;
40 #X connect 14 0 15 0;
#X connect 16 0 3 0;
#X connect 17 0 8 1;
#X connect 17 1 9 0;
#X connect 17 2 11 0;
45 #X connect 17 3 14 0;
#X connect 17 4 18 0;
#X connect 18 0 16 1;
#X connect 19 0 8 0;

```

27 prototype/mountains.pd

```

#N canvas 0 0 621 661 10;
#X obj 17 23 inlet;
#X obj 52 70 separator;
#X obj 51 150 t a a;
5 #X obj 329 247 GEMglBegin GL_QUADS;
#X obj 285 415 GEMglVertex2f;
#X obj 285 435 GEMglVertex2f;
#X obj 285 455 GEMglVertex2f;
#X obj 285 475 GEMglVertex2f;
10 #X obj 265 248 GEMglEnd;
#X obj 265 222 t a a a;
#X obj 285 277 t b a b;
#X msg 285 300 64;
#X obj 285 321 until;
15 #X obj 285 342 f;
#X obj 285 363 t b f;
#X msg 326 300 0;
#X obj 318 341 + 1;
#X obj 378 366 t f f;
20 #X obj 378 387 - 1;
#X obj 376 433 tabread \%0-height;
#X obj 377 457 tabread \%0-height;
#X obj 49 254 GEMglEnd;
#X obj 49 228 t a a a;
25 #X obj 65 435 GEMglVertex2f;
#X obj 65 455 GEMglVertex2f;
#X obj 65 277 t b a b;
#X msg 65 300 64;
#X obj 65 321 until;
30 #X obj 65 342 f;
#X obj 65 363 t b f;
#X obj 65 384 gemlist;
#X msg 106 300 0;
#X obj 98 341 + 1;
35 #X obj 158 366 t f f;
#X obj 158 387 - 1;

```

```

#X obj 156 433 tabread \%0-height;
#X obj 157 457 tabread \%0-height;
#X obj 113 253 GEMglBegin GL_LINES;
40 #X obj 52 111 scale 0.03125;
#X obj 52 132 translateXYZ -31 0 0;
#X obj 476 89 table \%0-height 64;
#X obj 300 19 loadbang;
#X msg 300 40 64;
45 #X obj 300 61 until;
#X obj 300 82 f 0;
#X obj 363 71 + 1;
#X obj 363 92 mod 64;
#X obj 300 103 t b f;
50 #X obj 300 124 random 64000;
#X obj 300 166 tabwrite \%0-height;
#X obj 51 200 translateXYZ 0 0 0;
#X obj 18 45 t a a;
#X obj 19 544 outlet;
55 #X obj 285 384 gemlist;
#X obj 261 201 colorRGB 0 0 0;
#X obj 302 145 / 2000;
#X obj 17 503 translateXYZ 0 0 -500;
#X obj 18 482 scaleXYZ 1 0.6 1;
60 #X obj 51 171 colorRGB 0.5 0.5 0.5;
#X connect 0 0 51 0;
#X connect 1 0 38 0;
#X connect 2 0 58 0;
#X connect 2 1 54 0;
65 #X connect 4 0 5 0;
#X connect 5 0 6 0;
#X connect 6 0 7 0;
#X connect 9 0 8 0;
#X connect 9 1 10 0;
70 #X connect 9 2 3 0;
#X connect 10 0 11 0;
#X connect 10 1 53 1;
#X connect 10 2 15 0;
#X connect 11 0 12 0;
75 #X connect 12 0 13 0;
#X connect 13 0 14 0;
#X connect 13 0 16 0;
#X connect 14 0 53 0;
#X connect 14 1 17 0;
80 #X connect 15 0 13 1;
#X connect 16 0 13 1;
#X connect 17 0 18 0;
#X connect 17 1 6 1;
#X connect 17 1 7 1;
85 #X connect 17 1 20 0;
#X connect 18 0 4 1;
#X connect 18 0 5 1;
#X connect 18 0 19 0;
#X connect 19 0 5 2;
90 #X connect 20 0 6 2;
#X connect 22 0 21 0;
#X connect 22 1 25 0;
#X connect 22 2 37 0;

```

```

#X connect 23 0 24 0;
95 #X connect 25 0 26 0;
#X connect 25 1 30 1;
#X connect 25 2 31 0;
#X connect 26 0 27 0;
#X connect 27 0 28 0;
100 #X connect 28 0 29 0;
#X connect 28 0 32 0;
#X connect 29 0 30 0;
#X connect 29 1 33 0;
#X connect 30 0 23 0;
105 #X connect 31 0 28 1;
#X connect 32 0 28 1;
#X connect 33 0 34 0;
#X connect 33 1 24 1;
#X connect 33 1 36 0;
110 #X connect 34 0 23 1;
#X connect 34 0 35 0;
#X connect 35 0 23 2;
#X connect 36 0 24 2;
#X connect 38 0 39 0;
115 #X connect 39 0 2 0;
#X connect 41 0 42 0;
#X connect 42 0 43 0;
#X connect 43 0 44 0;
#X connect 44 0 45 0;
120 #X connect 44 0 47 0;
#X connect 45 0 46 0;
#X connect 46 0 44 1;
#X connect 47 0 48 0;
#X connect 47 1 49 1;
125 #X connect 48 0 55 0;
#X connect 50 0 22 0;
#X connect 51 0 57 0;
#X connect 51 1 1 0;
#X connect 53 0 4 0;
130 #X connect 54 0 9 0;
#X connect 55 0 49 0;
#X connect 56 0 52 0;
#X connect 57 0 56 0;
#X connect 58 0 50 0;

```

28 prototype/mountainss.pd

```

#N canvas 95 296 450 413 10;
#X obj 82 126 mountains;
#X obj 81 10 inlet;
#X obj 81 31 separator;
5 #X obj 81 82 translateXYZ 0 -6 6000;
#X obj 81 103 scaleXYZ 7000 500 1;
#X obj 82 146 mountains;
#X obj 82 166 mountains;
#X obj 82 185 mountains;
10 #X obj 82 206 mountains;
#X obj 82 226 mountains;
#X obj 82 246 mountains;
#X obj 82 266 mountains;

```

```

#X obj 82 286 mountains;
15 #X obj 82 306 mountains;
#X obj 82 327 colorRGB 1 1 1;
#X obj 81 52 without-lighting;
#X connect 0 0 5 0;
#X connect 1 0 2 0;
20 #X connect 2 0 15 0;
#X connect 3 0 4 0;
#X connect 4 0 0 0;
#X connect 5 0 6 0;
#X connect 6 0 7 0;
25 #X connect 7 0 8 0;
#X connect 8 0 9 0;
#X connect 9 0 10 0;
#X connect 10 0 11 0;
#X connect 11 0 12 0;
30 #X connect 12 0 13 0;
#X connect 13 0 14 0;
#X connect 15 0 3 0;

```

29 prototype/multi-texture-1.pd

```

#N canvas 0 0 450 300 10;
#X obj 26 67 gemlist;
#X obj 26 109 pix_texture;
#X obj 108 43 loadbang;
5 #X msg 108 64 quality 2 \, repeat 0;
#X obj 26 88 pix_image \%2\%4\%3;
#X obj 96 21 r \%1-gemlist;
#X obj 26 130 s \%1-out;
#X obj 26 21 r \%1-index;
10 #X obj 26 42 sel \%4;
#X connect 0 0 4 0;
#X connect 1 0 6 0;
#X connect 2 0 3 0;
#X connect 3 0 1 0;
15 #X connect 4 0 1 0;
#X connect 5 0 0 1;
#X connect 7 0 8 0;
#X connect 8 0 0 0;

```

30 prototype/multi-texture.pd

```

#N canvas 0 0 450 300 10;
#X obj 24 16 inlet;
#X obj 24 235 outlet;
#X obj 163 18 inlet;
5 #X obj 24 41 t b a;
#X obj 24 62 f;
#X obj 23 112 multi-texture-1 \%0 \%1 \%2 0;
#X obj 57 64 s \%0-gemlist;
#X obj 26 210 r \%0-out;
10 #X obj 24 83 s \%0-index;
#X obj 23 132 multi-texture-1 \%0 \%1 \%2 1;
#X obj 23 152 multi-texture-1 \%0 \%1 \%2 2;
#X obj 23 172 multi-texture-1 \%0 \%1 \%2 3;

```

```

#X obj 23 192 multi-texture-1 \$0 \$1 \$2 4;
15 #X obj 193 112 multi-texture-1 \$0 \$1 \$2 5;
#X obj 193 132 multi-texture-1 \$0 \$1 \$2 6;
#X obj 193 152 multi-texture-1 \$0 \$1 \$2 7;
#X obj 193 172 multi-texture-1 \$0 \$1 \$2 8;
#X obj 193 192 multi-texture-1 \$0 \$1 \$2 9;
20 #X connect 0 0 3 0;
#X connect 2 0 4 1;
#X connect 3 0 4 0;
#X connect 3 1 6 0;
#X connect 4 0 8 0;
25 #X connect 7 0 1 0;

```

31 prototype/once.pd

```

#N canvas 0 0 450 300 10;
#X obj 28 27 inlet;
#X obj 28 48 spigot \$1;
#X obj 28 69 t a b;
5 #X obj 28 90 outlet;
#X msg 102 63 0;
#X obj 80 29 inlet;
#X connect 0 0 1 0;
#X connect 1 0 2 0;
10 #X connect 2 0 3 0;
#X connect 2 1 4 0;
#X connect 4 0 1 1;
#X connect 5 0 1 1;

```

32 prototype/repeat-gemlist.pd

```

#N canvas 0 0 450 300 10;
#X obj 78 63 t b a b;
#X obj 79 105 until;
#X obj 79 126 f 0;
5 #X obj 111 126 + 1;
#X msg 120 84 0;
#X obj 53 24 inlet;
#X obj 78 84 f \$1;
#X obj 79 147 t b f;
10 #X obj 79 168 gemlist;
#X obj 79 189 outlet;
#X obj 167 187 outlet;
#X obj 142 23 inlet;
#X connect 0 0 6 0;
15 #X connect 0 1 8 1;
#X connect 0 2 4 0;
#X connect 1 0 2 0;
#X connect 2 0 3 0;
#X connect 2 0 7 0;
20 #X connect 3 0 2 1;
#X connect 4 0 2 1;
#X connect 5 0 0 0;
#X connect 6 0 1 0;
#X connect 7 0 8 0;
25 #X connect 7 1 10 0;

```



```
#X connect 8 0 9 0;
#X connect 11 0 6 1;
```

33 prototype/snake.pd

```
#N canvas 0 0 392 319 10;
#X obj 32 27 inlet;
#X obj 113 258 outlet;
#X obj 285 31 inlet;
5 #X obj 180 262 outlet;
#X obj 128 224 geo-pipe;
#X obj 139 103 t f f;
#X obj 134 148 / 16;
#X obj 169 125 + 1;
10 #X obj 169 146 / 16;
#X obj 29 131 repeat-gemlist 16;
#X obj 33 48 t a b;
#X obj 34 69 separator;
#X obj 212 129 t b b b;
15 #X msg 297 116 0;
#X obj 34 111 t a a b;
#X obj 245 242 translate 1 0 0 0;
#X obj 246 221 separator;
#X obj 245 263 outlet;
20 #X obj 35 90 scale 4;
#X obj 287 185 snake-shape;
#X obj 129 184 snake-shape;
#X obj 206 185 snake-shape;
#X obj 351 28 inlet;
25 #X obj 315 269 outlet;
#X connect 0 0 10 0;
#X connect 2 0 19 2;
#X connect 2 0 20 2;
#X connect 2 0 21 2;
30 #X connect 4 0 1 0;
#X connect 4 1 3 0;
#X connect 5 0 6 0;
#X connect 5 1 7 0;
#X connect 6 0 20 0;
35 #X connect 7 0 8 0;
#X connect 8 0 21 0;
#X connect 9 0 4 0;
#X connect 9 1 5 0;
#X connect 10 0 11 0;
40 #X connect 10 1 12 0;
#X connect 11 0 18 0;
#X connect 12 0 20 1;
#X connect 12 1 21 1;
#X connect 12 2 19 1;
45 #X connect 13 0 19 0;
#X connect 14 0 9 0;
#X connect 14 1 16 0;
#X connect 14 2 13 0;
#X connect 15 0 17 0;
50 #X connect 16 0 15 0;
#X connect 18 0 14 0;
#X connect 19 0 15 2;
```

```

#X connect 20 0 4 1;
#X connect 21 0 4 2;
55 #X connect 21 0 23 0;
#X connect 22 0 19 3;
#X connect 22 0 21 3;
#X connect 22 0 20 3;

```

34 prototype/snake-shape.pd

```

#N canvas 0 0 706 438 10;
#X obj 76 25 inlet;
#X obj 20 23 inlet;
#X obj 40 383 outlet;
5 #X obj 23 140 *;
#X obj 81 111 swap 1;
#X obj 82 138 -;
#X obj 52 161 *;
#X obj 23 161 +;
10 #X obj 69 213 * 3.14159;
#X obj 68 191 * 2;
#X obj 69 234 cos;
#X obj 69 276 + 1;
#X obj 68 298 *;
15 #X obj 69 255 * -1;
#X obj 234 281 swap 1;
#X obj 234 302 -;
#X obj 145 19 inlet;
#X obj 145 40 route head tail;
20 #X obj 143 147 *;
#X obj 156 169 *;
#X obj 143 189 +;
#X obj 315 114 pack f f f;
#X obj 234 111 pack f f f;
25 #X msg 315 135 \$2 \$3;
#X obj 234 153 -;
#X obj 315 156 -;
#X obj 234 174 t f f;
#X obj 234 195 *;
30 #X obj 315 177 t f f;
#X obj 315 198 *;
#X obj 234 216 +;
#X obj 234 239 sqrt;
#X obj 234 260 - 1;
35 #X msg 234 132 \$2 \$3;
#X obj 40 355 pack f f f;
#X obj 20 74 t f f f f;
#X obj 145 61 unpack f f f;
#X obj 250 62 unpack f f f;
40 #X obj 68 319 +;
#X obj 183 147 *;
#X obj 196 169 *;
#X obj 183 189 +;
#X obj 20 113 t f f;
45 #X obj 316 218 +;
#X obj 76 46 t b b b;
#X obj 389 112 pack f f f;
#X msg 389 133 \$2 \$3;

```

```

#X obj 389 154 -;
50 #X obj 389 175 t f f;
#X obj 389 196 *;
#X obj 234 344 clip 0 1;
#X obj 519 25 inlet;
#X obj 234 323 / 4;
55 #X connect 0 0 44 0;
#X connect 1 0 35 0;
#X connect 3 0 7 0;
#X connect 4 0 5 0;
#X connect 4 1 5 1;
60 #X connect 5 0 6 0;
#X connect 5 0 19 0;
#X connect 5 0 40 0;
#X connect 6 0 7 1;
#X connect 7 0 34 0;
65 #X connect 8 0 10 0;
#X connect 9 0 8 0;
#X connect 10 0 13 0;
#X connect 11 0 12 0;
#X connect 12 0 38 0;
70 #X connect 13 0 11 0;
#X connect 14 0 15 0;
#X connect 14 1 15 1;
#X connect 15 0 52 0;
#X connect 16 0 17 0;
75 #X connect 17 0 36 0;
#X connect 17 1 37 0;
#X connect 18 0 20 0;
#X connect 19 0 20 1;
#X connect 20 0 34 2;
80 #X connect 21 0 23 0;
#X connect 22 0 33 0;
#X connect 23 0 25 0;
#X connect 24 0 26 0;
#X connect 25 0 28 0;
85 #X connect 26 0 27 0;
#X connect 26 1 27 1;
#X connect 27 0 30 0;
#X connect 28 0 29 0;
#X connect 28 1 29 1;
90 #X connect 29 0 43 0;
#X connect 30 0 31 0;
#X connect 31 0 32 0;
#X connect 32 0 14 0;
#X connect 33 0 24 0;
95 #X connect 34 0 2 0;
#X connect 35 0 42 0;
#X connect 35 1 18 0;
#X connect 35 1 39 0;
#X connect 35 2 4 0;
100 #X connect 36 0 3 1;
#X connect 36 0 22 1;
#X connect 36 1 18 1;
#X connect 36 1 21 1;
#X connect 36 2 39 1;
105 #X connect 36 2 45 1;
```

```

#X connect 37 0 6 1;
#X connect 37 0 22 2;
#X connect 37 1 19 1;
#X connect 37 1 21 2;
110 #X connect 37 2 40 1;
#X connect 37 2 45 2;
#X connect 38 0 34 1;
#X connect 39 0 41 0;
#X connect 40 0 41 1;
115 #X connect 41 0 38 1;
#X connect 42 0 3 0;
#X connect 42 1 9 0;
#X connect 43 0 30 1;
#X connect 44 0 22 0;
120 #X connect 44 1 21 0;
#X connect 44 2 45 0;
#X connect 45 0 46 0;
#X connect 46 0 47 0;
#X connect 47 0 48 0;
125 #X connect 48 0 49 0;
#X connect 48 1 49 1;
#X connect 49 0 43 1;
#X connect 50 0 12 1;
#X connect 51 0 50 2;
130 #X connect 52 0 50 0;

```

35 prototype/snakes.pd_lua

```

local S = pd.Class:new():register("snakes")

function S:initialize(name, atoms)
  self.snakes = { }
  5 self.inlets = 2
  self.outlets = 3
  self.behaviour = "wriggle"
  self.maxheight = 4
  self.other = { head = "tail", tail = "head" }
  10 return true
end

function S:in_l_bang()
  local i, s, sel, msg
  15 for i,s in ipairs(self.snakes) do
    self:outlet(2, "head", s.head)
    self:outlet(2, "tail", s.tail)
    self:outlet(1, "bang", {})
  end
  20 end

function S:get_target(head)
  local a = math.random() * 2.0 * math.pi
  local x = head[1] + 2 * math.cos(a)
  25 local y = head[2] + 2 * math.sin(a)
  if x * x + y * y < 40 * 40 then return { x, y, 0 } else return self:get_target(
    ↪ (head) end
end

```

```

function S:in_2_tick(atoms)
30   local dt = atoms[1]
   local i, s, a, f, dx, dy, dz
   for i,s in ipairs(self.snakes) do
       s.phase = s.phase + dt
       if s.phase > 1 then
35         s.phase = s.phase - 1 - 0.1 * math.random()
         self:outlet(3, "list", { i, s.target[1], s.target[2], s.target[3] })
         a = math.random() * 2.0 * math.pi
         if self.behaviour == "wriggle" or (self.behaviour == "column" and s.move[
           ↵ [1] == "head") then
             if s.move[1] == "tail" then
40               s.source[1] = s.head[1]
               s.source[2] = s.head[2]
               s.source[3] = s.head[3]
               s.target = { 0.5 * (s.tail[1] + s.head[1]), 0.5 * (s.tail[2] + s.head[
                 ↵ [2]), 0 }
             else
45               s.source[1] = s.tail[1]
               s.source[2] = s.tail[2]
               s.source[3] = s.tail[3]
               s.target = self:get_target(s.head)
             end
50             s.move[1] = self.other[s.move[1]]
             else if self.behaviour == "column" then
               s.source[1] = s.target[1]
               s.source[2] = s.target[2]
               s.source[3] = s.target[3]
55               s.move[1] = "tail"
               s.target[1] = s.head[1]
               s.target[2] = s.head[2]
               s.target[3] = math.min(s.target[3] + 0.25, self.maxheight)
             end end
60         else
           f = 1 - math.exp(-5 * math.max(s.phase, 0))
           s[s.move[1]][1] = s.source[1] * (1 - f) + f * s.target[1]
           s[s.move[1]][2] = s.source[2] * (1 - f) + f * s.target[2]
           s[s.move[1]][3] = s.source[3] * (1 - f) + f * s.target[3]
65         end
       end
     end

function S:in_2_reset(atoms)
70   local i, x, y, z, b, n, t, d
   if "number" == type(atoms[1]) then n = atoms[1] else n = 10 end
   r = math.sqrt(n)
   self.snakes = { }
   for i = 1,n do
75     a = math.random() * 2.0 * math.pi
     d = math.random() * 40
     x = d * math.cos(a)
     y = d * math.sin(a)
     z = 0
80     a = math.random() * 2.0 * math.pi
     b = math.random() * 2.0 * math.pi
     t = math.random()
     self.snakes[i] =

```

```

85      { head = { x, y, z }
        , tail = { x + math.cos(a), y + math.sin(a), z }
        , move = { "tail" }
        , source = { x + math.cos(a), y + math.sin(a), z }
        , target = { x + 2 * math.cos(b), y + 2 * math.sin(b), z }
        , phase = t
90    }
    end
end

function S:in_2_wriggle() self.behaviour = "wriggle" end
95 function S:in_2_column() self.behaviour = "column" end
function S:in_2_height(atoms) self.maxheight = 6 * atoms[1] end

```

36 prototype/starfield.pd

```

#N canvas 0 0 641 450 10;
#X obj 29 24 inlet;
#X obj 29 50 separator;
#X obj 30 94 scale 8000;
5 #X obj 249 12 loadbang;
#X obj 286 109 until;
#X obj 286 133 t b b b;
#X obj 286 159 random 1e+06;
#X obj 286 180 / 1e+06;
10 #X obj 286 201 - 0.5;
#X obj 371 159 random 1e+06;
#X obj 371 180 / 1e+06;
#X obj 456 159 random 1e+06;
#X obj 456 180 / 1e+06;
15 #X obj 286 230 pack f f f;
#X obj 285 251 vec3-normalize;
#X obj 286 83 t f b;
#X msg 345 95 0;
#X obj 381 312 + 1;
20 #X obj 283 320 t a b;
#X obj 232 342 unpack f f f;
#X obj 232 363 tabwrite \ $0-vx;
#X obj 285 412 tabwrite \ $0-vz;
#X obj 256 384 tabwrite \ $0-vy;
25 #X obj 330 314 f 0;
#X obj 247 66 t b f;
#X obj 247 87 f \ $0;
#X obj 55 196 gemvertexbuffer 20000;
#X msg 30 162 position \ $1-vx \ $1-vy \ $1-vz \, draw lines;
30 #X msg 248 47 20000;
#X obj 429 282 table \ $0-vx 20000;
#X obj 429 303 table \ $0-vy 20000;
#X obj 429 324 table \ $0-vz 20000;
#X obj 286 271 t a a;
35 #X obj 29 71 without-lighting;
#X obj 174 299 vec3-add 0 0.001 0;
#X obj 456 201 - 0.5;
#X obj 371 201 * 0.5;
#X obj 249 28 t b b;
40 #X connect 0 0 1 0;
#X connect 1 0 33 0;

```

```

#X connect 2 0 26 0;
#X connect 3 0 37 0;
#X connect 4 0 5 0;
45 #X connect 5 0 6 0;
#X connect 5 1 9 0;
#X connect 5 2 11 0;
#X connect 6 0 7 0;
#X connect 7 0 8 0;
50 #X connect 8 0 13 0;
#X connect 9 0 10 0;
#X connect 10 0 36 0;
#X connect 11 0 12 0;
#X connect 12 0 35 0;
55 #X connect 13 0 14 0;
#X connect 14 0 32 0;
#X connect 15 0 4 0;
#X connect 15 1 16 0;
#X connect 16 0 23 1;
60 #X connect 17 0 23 1;
#X connect 18 0 19 0;
#X connect 18 1 23 0;
#X connect 19 0 20 0;
#X connect 19 1 22 0;
65 #X connect 19 2 21 0;
#X connect 23 0 20 1;
#X connect 23 0 22 1;
#X connect 23 0 21 1;
#X connect 23 0 17 0;
70 #X connect 24 0 25 0;
#X connect 24 1 15 0;
#X connect 25 0 27 0;
#X connect 27 0 26 0;
#X connect 28 0 24 0;
75 #X connect 32 0 34 0;
#X connect 32 1 18 0;
#X connect 33 0 2 0;
#X connect 34 0 18 0;
#X connect 35 0 13 2;
80 #X connect 36 0 13 1;
#X connect 37 0 28 0;
#X connect 37 1 28 0;

```

37 prototype/start.sh

```

#!/bin/bash
pd \
-nrt -r 48000 -channels 2 \
-lib pdlua:Gem \
5 -open anaglyph-help.pd \
-open ../sketches/thunder/thunder4.pd

```

38 prototype/teapot.obj

```

# Max2Obj Version 4.0 Mar 10th, 2001
#
# object Teapot01 to come ...

```

```

#
5  v  5.929688  4.125000  0.000000
   v  5.832031  4.494141  0.000000
   v  5.945313  4.617188  0.000000
   v  6.175781  4.494141  0.000000
   v  6.429688  4.125000  0.000000
10  v  5.387188  4.125000  2.747500
   v  5.297100  4.494141  2.709170
   v  5.401602  4.617188  2.753633
   v  5.614209  4.494141  2.844092
   v  5.848437  4.125000  2.943750
15  v  3.899688  4.125000  4.970000
   v  3.830352  4.494141  4.900664
   v  3.910782  4.617188  4.981094
   v  4.074414  4.494141  5.144727
   v  4.254687  4.125000  5.325000
20  v  1.677188  4.125000  6.457500
   v  1.638858  4.494141  6.367412
   v  1.683320  4.617188  6.471914
   v  1.773780  4.494141  6.684522
   v  1.873438  4.125000  6.918750
25  v  -1.070312  4.125000  7.000000
   v  -1.070312  4.494141  6.902344
   v  -1.070312  4.617188  7.015625
   v  -1.070312  4.494141  7.246094
   v  -1.070312  4.125000  7.500000
30  v  -4.007656  4.125000  6.457500
   v  -3.859572  4.494141  6.367412
   v  -3.847676  4.617188  6.471914
   v  -3.917371  4.494141  6.684522
   v  -4.014062  4.125000  6.918750
35  v  -6.209063  4.125000  4.970000
   v  -6.042168  4.494141  4.900664
   v  -6.072500  4.617188  4.981094
   v  -6.217675  4.494141  5.144727
   v  -6.395312  4.125000  5.325000
40  v  -7.591093  4.125000  2.747500
   v  -7.464421  4.494141  2.709170
   v  -7.550137  4.617188  2.753633
   v  -7.755822  4.494141  2.844092
   v  -7.989062  4.125000  2.943750
45  v  -8.070313  4.125000  0.000000
   v  -7.972656  4.494141  0.000000
   v  -8.085938  4.617188  0.000000
   v  -8.316406  4.494141  0.000000
   v  -8.570313  4.125000  0.000000
50  v  -7.527812  4.125000  -2.747500
   v  -7.437724  4.494141  -2.709170
   v  -7.542227  4.617188  -2.753633
   v  -7.754834  4.494141  -2.844092
   v  -7.989062  4.125000  -2.943750
55  v  -6.040312  4.125000  -4.970000
   v  -5.970977  4.494141  -4.900664
   v  -6.051406  4.617188  -4.981094
   v  -6.215039  4.494141  -5.144727
   v  -6.395312  4.125000  -5.325000
60  v  -3.817812  4.125000  -6.457500

```


	v	-3.779482	4.494141	-6.367412
	v	-3.823945	4.617188	-6.471914
	v	-3.914404	4.494141	-6.684522
	v	-4.014062	4.125000	-6.918750
65	v	-1.070312	4.125000	-7.000000
	v	-1.070312	4.494141	-6.902344
	v	-1.070312	4.617188	-7.015625
	v	-1.070312	4.494141	-7.246094
	v	-1.070312	4.125000	-7.500000
70	v	1.677188	4.125000	-6.457500
	v	1.638858	4.494141	-6.367412
	v	1.683320	4.617188	-6.471914
	v	1.773780	4.494141	-6.684522
	v	1.873438	4.125000	-6.918750
75	v	3.899688	4.125000	-4.970000
	v	3.830352	4.494141	-4.900664
	v	3.910782	4.617188	-4.981094
	v	4.074414	4.494141	-5.144727
	v	4.254687	4.125000	-5.325000
80	v	5.387188	4.125000	-2.747500
	v	5.297100	4.494141	-2.709170
	v	5.401602	4.617188	-2.753633
	v	5.614209	4.494141	-2.844092
	v	5.848437	4.125000	-2.943750
85	v	7.347656	2.162109	0.000000
	v	8.148438	0.234375	0.000000
	v	8.714844	-1.623047	0.000000
	v	8.929688	-3.375000	0.000000
	v	6.695264	2.162109	3.304053
90	v	7.433985	0.234375	3.618360
	v	7.956494	-1.623047	3.840674
	v	8.154688	-3.375000	3.925000
	v	4.906446	2.162109	5.976758
	v	5.475000	0.234375	6.545312
95	v	5.877149	-1.623047	6.947461
	v	6.029688	-3.375000	7.100000
	v	2.233740	2.162109	7.765576
	v	2.548047	0.234375	8.504297
	v	2.770362	-1.623047	9.026807
100	v	2.854688	-3.375000	9.225000
	v	-1.070312	2.162109	8.417969
	v	-1.070312	0.234375	9.218750
	v	-1.070312	-1.623047	9.785156
	v	-1.070312	-3.375000	10.000000
105	v	-4.374365	2.162109	7.765576
	v	-4.688672	0.234375	8.504297
	v	-4.910986	-1.623047	9.026807
	v	-4.995313	-3.375000	9.225000
	v	-7.047071	2.162109	5.976758
110	v	-7.615624	0.234375	6.545312
	v	-8.017773	-1.623047	6.947461
	v	-8.170312	-3.375000	7.100000
	v	-8.835889	2.162109	3.304053
	v	-9.574610	0.234375	3.618360
115	v	-10.097119	-1.623047	3.840674
	v	-10.295313	-3.375000	3.925000
	v	-9.488281	2.162109	0.000000

	v	-10.289063	0.234375	0.000000
	v	-10.855469	-1.623047	0.000000
120	v	-11.070313	-3.375000	0.000000
	v	-8.835889	2.162109	-3.304053
	v	-9.574610	0.234375	-3.618360
	v	-10.097119	-1.623047	-3.840674
	v	-10.295313	-3.375000	-3.925000
125	v	-7.047071	2.162109	-5.976758
	v	-7.615624	0.234375	-6.545312
	v	-8.017773	-1.623047	-6.947461
	v	-8.170312	-3.375000	-7.100000
	v	-4.374365	2.162109	-7.765576
130	v	-4.688672	0.234375	-8.504297
	v	-4.910986	-1.623047	-9.026807
	v	-4.995313	-3.375000	-9.225000
	v	-1.070312	2.162109	-8.417969
	v	-1.070312	0.234375	-9.218750
135	v	-1.070312	-1.623047	-9.785156
	v	-1.070312	-3.375000	-10.000000
	v	2.233740	2.162109	-7.765576
	v	2.548047	0.234375	-8.504297
	v	2.770362	-1.623047	-9.026807
140	v	2.854688	-3.375000	-9.225000
	v	4.906446	2.162109	-5.976758
	v	5.475000	0.234375	-6.545312
	v	5.877149	-1.623047	-6.947461
	v	6.029688	-3.375000	-7.100000
145	v	6.695264	2.162109	-3.304053
	v	7.433985	0.234375	-3.618360
	v	7.956494	-1.623047	-3.840674
	v	8.154688	-3.375000	-3.925000
	v	8.539063	-4.857422	0.000000
150	v	7.679688	-5.953125	0.000000
	v	6.820313	-6.697266	0.000000
	v	6.429688	-7.125000	0.000000
	v	7.794336	-4.857422	3.771680
	v	7.001562	-5.953125	3.434375
155	v	6.208789	-6.697266	3.097070
	v	5.848437	-7.125000	2.943750
	v	5.752343	-4.857422	6.822656
	v	5.142187	-5.953125	6.212500
	v	4.532031	-6.697266	5.602344
160	v	4.254687	-7.125000	5.325000
	v	2.701367	-4.857422	8.864649
	v	2.364063	-5.953125	8.071875
	v	2.026758	-6.697266	7.279101
	v	1.873438	-7.125000	6.918750
165	v	-1.070312	-4.857422	9.609375
	v	-1.070312	-5.953125	8.750000
	v	-1.070312	-6.697266	7.890625
	v	-1.070312	-7.125000	7.500000
	v	-4.841992	-4.857422	8.864649
170	v	-4.504687	-5.953125	8.071875
	v	-4.167383	-6.697266	7.279101
	v	-4.014062	-7.125000	6.918750
	v	-7.892968	-4.857422	6.822656
	v	-7.282812	-5.953125	6.212500

175	v	-6.672656	-6.697266	5.602344
	v	-6.395312	-7.125000	5.325000
	v	-9.934961	-4.857422	3.771680
	v	-9.142187	-5.953125	3.434375
	v	-8.349414	-6.697266	3.097070
180	v	-7.989062	-7.125000	2.943750
	v	-10.679688	-4.857422	0.000000
	v	-9.820313	-5.953125	0.000000
	v	-8.960938	-6.697266	0.000000
	v	-8.570313	-7.125000	0.000000
185	v	-9.934961	-4.857422	-3.771680
	v	-9.142187	-5.953125	-3.434375
	v	-8.349414	-6.697266	-3.097070
	v	-7.989062	-7.125000	-2.943750
	v	-7.892968	-4.857422	-6.822656
190	v	-7.282812	-5.953125	-6.212500
	v	-6.672656	-6.697266	-5.602344
	v	-6.395312	-7.125000	-5.325000
	v	-4.841992	-4.857422	-8.864649
	v	-4.504687	-5.953125	-8.071875
195	v	-4.167383	-6.697266	-7.279101
	v	-4.014062	-7.125000	-6.918750
	v	-1.070312	-4.857422	-9.609375
	v	-1.070312	-5.953125	-8.750000
	v	-1.070312	-6.697266	-7.890625
200	v	-1.070312	-7.125000	-7.500000
	v	2.701367	-4.857422	-8.864649
	v	2.364063	-5.953125	-8.071875
	v	2.026758	-6.697266	-7.279101
	v	1.873438	-7.125000	-6.918750
205	v	5.752343	-4.857422	-6.822656
	v	5.142187	-5.953125	-6.212500
	v	4.532031	-6.697266	-5.602344
	v	4.254687	-7.125000	-5.325000
	v	7.794336	-4.857422	-3.771680
210	v	7.001562	-5.953125	-3.434375
	v	6.208789	-6.697266	-3.097070
	v	5.848437	-7.125000	-2.943750
	v	6.259766	-7.400391	0.000000
	v	5.351563	-7.640625	0.000000
215	v	3.107422	-7.810547	0.000000
	v	-1.070312	-7.875000	0.000000
	v	5.691685	-7.400391	2.877056
	v	4.853868	-7.640625	2.520586
	v	2.783648	-7.810547	1.639761
220	v	4.134043	-7.400391	5.204355
	v	3.489219	-7.640625	4.559531
	v	1.895879	-7.810547	2.966191
	v	1.806743	-7.400391	6.761997
	v	1.450274	-7.640625	5.924180
225	v	0.569448	-7.810547	3.853960
	v	-1.070312	-7.400391	7.330078
	v	-1.070312	-7.640625	6.421875
	v	-1.070312	-7.810547	4.177734
	v	-3.947368	-7.400391	6.761997
230	v	-3.590898	-7.640625	5.924180
	v	-2.710073	-7.810547	3.853960

	v	-6.274668	-7.400391	5.204355
	v	-5.629844	-7.640625	4.559531
	v	-4.036504	-7.810547	2.966191
235	v	-7.832309	-7.400391	2.877056
	v	-6.994492	-7.640625	2.520586
	v	-4.924272	-7.810547	1.639761
	v	-8.400391	-7.400391	0.000000
	v	-7.492188	-7.640625	0.000000
240	v	-5.248047	-7.810547	0.000000
	v	-7.832309	-7.400391	-2.877056
	v	-6.994492	-7.640625	-2.520586
	v	-4.924272	-7.810547	-1.639761
	v	-6.274668	-7.400391	-5.204355
245	v	-5.629844	-7.640625	-4.559531
	v	-4.036504	-7.810547	-2.966191
	v	-3.947368	-7.400391	-6.761997
	v	-3.590898	-7.640625	-5.924180
	v	-2.710073	-7.810547	-3.853960
250	v	-1.070312	-7.400391	-7.330078
	v	-1.070312	-7.640625	-6.421875
	v	-1.070312	-7.810547	-4.177734
	v	1.806743	-7.400391	-6.761997
	v	1.450274	-7.640625	-5.924180
255	v	0.569448	-7.810547	-3.853960
	v	4.134043	-7.400391	-5.204355
	v	3.489219	-7.640625	-4.559531
	v	1.895879	-7.810547	-2.966191
	v	5.691685	-7.400391	-2.877056
260	v	4.853868	-7.640625	-2.520586
	v	2.783648	-7.810547	-1.639761
	v	-9.070313	2.250000	0.000000
	v	-11.406250	2.232422	0.000000
	v	-13.132813	2.109375	0.000000
265	v	-14.203125	1.775391	0.000000
	v	-14.570313	1.125000	0.000000
	v	-8.992188	2.425781	0.843750
	v	-11.475830	2.405457	0.843750
	v	-13.298828	2.263184	0.843750
270	v	-14.421631	1.877014	0.843750
	v	-14.804688	1.125000	0.843750
	v	-8.820313	2.812500	1.125000
	v	-11.628906	2.786134	1.125000
	v	-13.664063	2.601563	1.125000
275	v	-14.902344	2.100586	1.125000
	v	-15.320313	1.125000	1.125000
	v	-8.648438	3.199219	0.843750
	v	-11.781982	3.166809	0.843750
	v	-14.029297	2.939941	0.843750
280	v	-15.383057	2.324158	0.843750
	v	-15.835938	1.125000	0.843750
	v	-8.570313	3.375000	0.000000
	v	-11.851563	3.339844	0.000000
	v	-14.195313	3.093750	0.000000
285	v	-15.601563	2.425781	0.000000
	v	-16.070313	1.125000	0.000000
	v	-8.648438	3.199219	-0.843750
	v	-11.781982	3.166809	-0.843750

```

v -14.029297 2.939941 -0.843750
290 v -15.383057 2.324158 -0.843750
v -15.835938 1.125000 -0.843750
v -8.820313 2.812500 -1.125000
v -11.628906 2.786134 -1.125000
v -13.664063 2.601563 -1.125000
295 v -14.902344 2.100586 -1.125000
v -15.320313 1.125000 -1.125000
v -8.992188 2.425781 -0.843750
v -11.475830 2.405457 -0.843750
v -13.298828 2.263184 -0.843750
300 v -14.421631 1.877014 -0.843750
v -14.804688 1.125000 -0.843750
v -14.375000 0.105469 0.000000
v -13.757813 -1.125000 0.000000
v -12.671875 -2.355469 0.000000
305 v -11.070313 -3.375000 0.000000
v -14.588013 0.007050 0.843750
v -13.909180 -1.275146 0.843750
v -12.724976 -2.540863 0.843750
v -10.992188 -3.609375 0.843750
310 v -15.056641 -0.209473 1.125000
v -14.242188 -1.605469 1.125000
v -12.841797 -2.948730 1.125000
v -10.820313 -4.125000 1.125000
v -15.525269 -0.425995 0.843750
315 v -14.575195 -1.935791 0.843750
v -12.958618 -3.356598 0.843750
v -10.648438 -4.640625 0.843750
v -15.738281 -0.524414 0.000000
v -14.726563 -2.085938 0.000000
320 v -13.011719 -3.541992 0.000000
v -10.570313 -4.875000 0.000000
v -15.525269 -0.425995 -0.843750
v -14.575195 -1.935791 -0.843750
v -12.958618 -3.356598 -0.843750
325 v -10.648438 -4.640625 -0.843750
v -15.056641 -0.209473 -1.125000
v -14.242188 -1.605469 -1.125000
v -12.841797 -2.948730 -1.125000
v -10.820313 -4.125000 -1.125000
330 v -14.588013 0.007050 -0.843750
v -13.909180 -1.275146 -0.843750
v -12.724976 -2.540863 -0.843750
v -10.992188 -3.609375 -0.843750
v 7.429688 -0.750000 0.000000
335 v 9.828125 -0.199219 0.000000
v 10.867188 1.125000 0.000000
v 11.437500 2.730469 0.000000
v 12.429688 4.125000 0.000000
v 7.429688 -1.394531 1.856250
340 v 10.011230 -0.677124 1.676074
v 11.101563 0.846680 1.279688
v 11.723145 2.629761 0.883301
v 12.898438 4.125000 0.703125
v 7.429688 -2.812500 2.475000
345 v 10.414063 -1.728516 2.234766

```

```

v 11.617188 0.234375 1.706250
v 12.351563 2.408203 1.177734
v 13.929688 4.125000 0.937500
v 7.429688 -4.230469 1.856250
350 v 10.816895 -2.779907 1.676074
v 12.132813 -0.377930 1.279688
v 12.979980 2.186646 0.883301
v 14.960938 4.125000 0.703125
v 7.429688 -4.875000 0.000000
355 v 11.000000 -3.257813 0.000000
v 12.367188 -0.656250 0.000000
v 13.265625 2.085938 0.000000
v 15.429688 4.125000 0.000000
v 7.429688 -4.230469 -1.856250
360 v 10.816895 -2.779907 -1.676074
v 12.132813 -0.377930 -1.279688
v 12.979980 2.186646 -0.883301
v 14.960938 4.125000 -0.703125
v 7.429688 -2.812500 -2.475000
365 v 10.414063 -1.728516 -2.234766
v 11.617188 0.234375 -1.706250
v 12.351563 2.408203 -1.177734
v 13.929688 4.125000 -0.937500
v 7.429688 -1.394531 -1.856250
370 v 10.011230 -0.677124 -1.676074
v 11.101563 0.846680 -1.279688
v 11.723145 2.629761 -0.883301
v 12.898438 4.125000 -0.703125
v 12.789063 4.335938 0.000000
375 v 13.054688 4.406250 0.000000
v 13.132813 4.335938 0.000000
v 12.929688 4.125000 0.000000
v 13.291077 4.346237 0.659180
v 13.525879 4.422729 0.562500
380 v 13.532898 4.350357 0.465820
v 13.242188 4.125000 0.421875
v 14.395508 4.368896 0.878906
v 14.562500 4.458984 0.750000
v 14.413086 4.382080 0.621094
385 v 13.929688 4.125000 0.562500
v 15.499939 4.391556 0.659180
v 15.599121 4.495239 0.562500
v 15.293274 4.413804 0.465820
v 14.617188 4.125000 0.421875
390 v 16.001953 4.401855 0.000000
v 16.070313 4.511719 0.000000
v 15.693359 4.428224 0.000000
v 14.929688 4.125000 0.000000
v 15.499939 4.391556 -0.659180
395 v 15.599121 4.495239 -0.562500
v 15.293274 4.413804 -0.465820
v 14.617188 4.125000 -0.421875
v 14.395508 4.368896 -0.878906
v 14.562500 4.458984 -0.750000
400 v 14.413086 4.382080 -0.621094
v 13.929688 4.125000 -0.562500
v 13.291077 4.346237 -0.659180

```

```

v 13.525879 4.422729 -0.562500
v 13.532898 4.350357 -0.465820
405 v 13.242188 4.125000 -0.421875
v -1.070312 7.875000 0.000000
v 0.632813 7.628906 0.000000
v 0.554688 7.031250 0.000000
v -0.085937 6.292969 0.000000
410 v -0.070312 5.625000 0.000000
v 0.501414 7.628906 0.670256
v 0.429278 7.031250 0.639395
v -0.162029 6.292969 0.386960
v -0.147812 5.625000 0.392500
415 v 0.140489 7.628906 1.210801
v 0.084844 7.031250 1.155156
v -0.370879 6.292969 0.699434
v -0.360312 5.625000 0.710000
v -0.400056 7.628906 1.571726
420 v -0.430918 7.031250 1.499590
v -0.683352 6.292969 0.908284
v -0.677812 5.625000 0.922500
v -1.070312 7.628906 1.703125
v -1.070312 7.031250 1.625000
425 v -1.070312 6.292969 0.984375
v -1.070312 5.625000 1.000000
v -1.740569 7.628906 1.571726
v -1.709707 7.031250 1.499590
v -1.457273 6.292969 0.908284
430 v -1.462812 5.625000 0.922500
v -2.281113 7.628906 1.210801
v -2.225469 7.031250 1.155156
v -1.769746 6.292969 0.699434
v -1.780312 5.625000 0.710000
435 v -2.642038 7.628906 0.670256
v -2.569902 7.031250 0.639395
v -1.978596 6.292969 0.386960
v -1.992812 5.625000 0.392500
v -2.773438 7.628906 0.000000
440 v -2.695313 7.031250 0.000000
v -2.054687 6.292969 0.000000
v -2.070312 5.625000 0.000000
v -2.642038 7.628906 -0.670256
v -2.569902 7.031250 -0.639395
445 v -1.978596 6.292969 -0.386960
v -1.992812 5.625000 -0.392500
v -2.281113 7.628906 -1.210801
v -2.225469 7.031250 -1.155156
v -1.769746 6.292969 -0.699434
450 v -1.780312 5.625000 -0.710000
v -1.740569 7.628906 -1.571726
v -1.709707 7.031250 -1.499590
v -1.457273 6.292969 -0.908284
v -1.462812 5.625000 -0.922500
455 v -1.070312 7.628906 -1.703125
v -1.070312 7.031250 -1.625000
v -1.070312 6.292969 -0.984375
v -1.070312 5.625000 -1.000000
v -0.400056 7.628906 -1.571726

```

```
460 v -0.430918 7.031250 -1.499590
v -0.683352 6.292969 -0.908284
v -0.677812 5.625000 -0.922500
v 0.140489 7.628906 -1.210801
v 0.084844 7.031250 -1.155156
465 v -0.370879 6.292969 -0.699434
v -0.360312 5.625000 -0.710000
v 0.501414 7.628906 -0.670256
v 0.429278 7.031250 -0.639395
v -0.162029 6.292969 -0.386960
470 v -0.147812 5.625000 -0.392500
v 1.210938 5.179688 0.000000
v 3.054688 4.875000 0.000000
v 4.710938 4.570313 0.000000
v 5.429688 4.125000 0.000000
475 v 1.034141 5.179688 0.895391
v 2.735000 4.875000 1.619062
v 4.262891 4.570313 2.269140
v 4.925938 4.125000 2.551250
v 0.549375 5.179688 1.619688
480 v 1.858438 4.875000 2.928750
v 3.034375 4.570313 4.104687
v 3.544688 4.125000 4.615000
v -0.174922 5.179688 2.104453
v 0.548750 4.875000 3.805313
485 v 1.198828 4.570313 5.333203
v 1.480938 4.125000 5.996250
v -1.070312 5.179688 2.281250
v -1.070312 4.875000 4.125000
v -1.070312 4.570313 5.781250
490 v -1.070312 4.125000 6.500000
v -1.965703 5.179688 2.104453
v -2.689375 4.875000 3.805313
v -3.339453 4.570313 5.333203
v -3.621562 4.125000 5.996250
495 v -2.690000 5.179688 1.619688
v -3.999062 4.875000 2.928750
v -5.174999 4.570313 4.104687
v -5.685312 4.125000 4.615000
v -3.174765 5.179688 0.895391
500 v -4.875625 4.875000 1.619062
v -6.403516 4.570313 2.269140
v -7.066563 4.125000 2.551250
v -3.351562 5.179688 0.000000
v -5.195313 4.875000 0.000000
505 v -6.851563 4.570313 0.000000
v -7.570313 4.125000 0.000000
v -3.174765 5.179688 -0.895391
v -4.875625 4.875000 -1.619062
v -6.403516 4.570313 -2.269140
510 v -7.066563 4.125000 -2.551250
v -2.690000 5.179688 -1.619688
v -3.999062 4.875000 -2.928750
v -5.174999 4.570313 -4.104687
v -5.685312 4.125000 -4.615000
515 v -1.965703 5.179688 -2.104453
v -2.689375 4.875000 -3.805313
```



```

v   -3.339453  4.570313  -5.333203
v   -3.621562  4.125000  -5.996250
v   -1.070312  5.179688  -2.281250
520 v   -1.070312  4.875000  -4.125000
v   -1.070312  4.570313  -5.781250
v   -1.070312  4.125000  -6.500000
v   -0.174922  5.179688  -2.104453
v    0.548750  4.875000  -3.805313
525 v    1.198828  4.570313  -5.333203
v    1.480938  4.125000  -5.996250
v    0.549375  5.179688  -1.619688
v    1.858438  4.875000  -2.928750
v    3.034375  4.570313  -4.104687
530 v    3.544688  4.125000  -4.615000
v    1.034141  5.179688  -0.895391
v    2.735000  4.875000  -1.619062
v    4.262891  4.570313  -2.269140
v    4.925938  4.125000  -2.551250
535 # 530 vertices

vt   2.000000  2.000000  0.000000
vt   2.000000  1.975000  0.000000
vt   2.000000  1.950000  0.000000
540 vt   2.000000  1.925000  0.000000
vt   2.000000  1.900000  0.000000
vt   1.750000  2.000000  0.000000
vt   1.750000  1.975000  0.000000
vt   1.750000  1.950000  0.000000
545 vt   1.750000  1.925000  0.000000
vt   1.750000  1.900000  0.000000
vt   1.500000  2.000000  0.000000
vt   1.500000  1.975000  0.000000
vt   1.500000  1.950000  0.000000
550 vt   1.500000  1.925000  0.000000
vt   1.500000  1.900000  0.000000
vt   1.250000  2.000000  0.000000
vt   1.250000  1.975000  0.000000
vt   1.250000  1.950000  0.000000
555 vt   1.250000  1.925000  0.000000
vt   1.250000  1.900000  0.000000
vt   1.000000  2.000000  0.000000
vt   1.000000  1.975000  0.000000
vt   1.000000  1.950000  0.000000
560 vt   1.000000  1.925000  0.000000
vt   1.000000  1.900000  0.000000
vt   1.000000  2.000000  0.000000
vt   1.000000  1.975000  0.000000
vt   1.000000  1.950000  0.000000
565 vt   1.000000  1.925000  0.000000
vt   1.000000  1.900000  0.000000
vt   0.750000  2.000000  0.000000
vt   0.750000  1.975000  0.000000
vt   0.750000  1.950000  0.000000
570 vt   0.750000  1.925000  0.000000
vt   0.750000  1.900000  0.000000
vt   0.500000  2.000000  0.000000
vt   0.500000  1.975000  0.000000

```

	vt	0.500000	1.950000	0.000000
575	vt	0.500000	1.925000	0.000000
	vt	0.500000	1.900000	0.000000
	vt	0.250000	2.000000	0.000000
	vt	0.250000	1.975000	0.000000
	vt	0.250000	1.950000	0.000000
580	vt	0.250000	1.925000	0.000000
	vt	0.250000	1.900000	0.000000
	vt	0.000000	2.000000	0.000000
	vt	0.000000	1.975000	0.000000
	vt	0.000000	1.950000	0.000000
585	vt	0.000000	1.925000	0.000000
	vt	0.000000	1.900000	0.000000
	vt	2.000000	2.000000	0.000000
	vt	2.000000	1.975000	0.000000
	vt	2.000000	1.950000	0.000000
590	vt	2.000000	1.925000	0.000000
	vt	2.000000	1.900000	0.000000
	vt	1.750000	2.000000	0.000000
	vt	1.750000	1.975000	0.000000
	vt	1.750000	1.950000	0.000000
595	vt	1.750000	1.925000	0.000000
	vt	1.750000	1.900000	0.000000
	vt	1.500000	2.000000	0.000000
	vt	1.500000	1.975000	0.000000
	vt	1.500000	1.950000	0.000000
600	vt	1.500000	1.925000	0.000000
	vt	1.500000	1.900000	0.000000
	vt	1.250000	2.000000	0.000000
	vt	1.250000	1.975000	0.000000
	vt	1.250000	1.950000	0.000000
605	vt	1.250000	1.925000	0.000000
	vt	1.250000	1.900000	0.000000
	vt	1.000000	2.000000	0.000000
	vt	1.000000	1.975000	0.000000
	vt	1.000000	1.950000	0.000000
610	vt	1.000000	1.925000	0.000000
	vt	1.000000	1.900000	0.000000
	vt	1.000000	2.000000	0.000000
	vt	1.000000	1.975000	0.000000
	vt	1.000000	1.950000	0.000000
615	vt	1.000000	1.925000	0.000000
	vt	1.000000	1.900000	0.000000
	vt	0.750000	2.000000	0.000000
	vt	0.750000	1.975000	0.000000
	vt	0.750000	1.950000	0.000000
620	vt	0.750000	1.925000	0.000000
	vt	0.750000	1.900000	0.000000
	vt	0.500000	2.000000	0.000000
	vt	0.500000	1.975000	0.000000
	vt	0.500000	1.950000	0.000000
625	vt	0.500000	1.925000	0.000000
	vt	0.500000	1.900000	0.000000
	vt	0.250000	2.000000	0.000000
	vt	0.250000	1.975000	0.000000
	vt	0.250000	1.950000	0.000000
630	vt	0.250000	1.925000	0.000000

	vt	0.250000	1.900000	0.000000
	vt	0.000000	2.000000	0.000000
	vt	0.000000	1.975000	0.000000
	vt	0.000000	1.950000	0.000000
635	vt	0.000000	1.925000	0.000000
	vt	0.000000	1.900000	0.000000
	vt	2.000000	1.900000	0.000000
	vt	2.000000	1.675000	0.000000
	vt	2.000000	1.450000	0.000000
640	vt	2.000000	1.225000	0.000000
	vt	2.000000	1.000000	0.000000
	vt	1.750000	1.900000	0.000000
	vt	1.750000	1.675000	0.000000
	vt	1.750000	1.450000	0.000000
645	vt	1.750000	1.225000	0.000000
	vt	1.750000	1.000000	0.000000
	vt	1.500000	1.900000	0.000000
	vt	1.500000	1.675000	0.000000
	vt	1.500000	1.450000	0.000000
650	vt	1.500000	1.225000	0.000000
	vt	1.500000	1.000000	0.000000
	vt	1.250000	1.900000	0.000000
	vt	1.250000	1.675000	0.000000
	vt	1.250000	1.450000	0.000000
655	vt	1.250000	1.225000	0.000000
	vt	1.250000	1.000000	0.000000
	vt	1.000000	1.900000	0.000000
	vt	1.000000	1.675000	0.000000
	vt	1.000000	1.450000	0.000000
660	vt	1.000000	1.225000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	1.000000	1.900000	0.000000
	vt	1.000000	1.675000	0.000000
	vt	1.000000	1.450000	0.000000
665	vt	1.000000	1.225000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	0.750000	1.900000	0.000000
	vt	0.750000	1.675000	0.000000
	vt	0.750000	1.450000	0.000000
670	vt	0.750000	1.225000	0.000000
	vt	0.750000	1.000000	0.000000
	vt	0.500000	1.900000	0.000000
	vt	0.500000	1.675000	0.000000
	vt	0.500000	1.450000	0.000000
675	vt	0.500000	1.225000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.250000	1.900000	0.000000
	vt	0.250000	1.675000	0.000000
	vt	0.250000	1.450000	0.000000
680	vt	0.250000	1.225000	0.000000
	vt	0.250000	1.000000	0.000000
	vt	0.000000	1.900000	0.000000
	vt	0.000000	1.675000	0.000000
	vt	0.000000	1.450000	0.000000
685	vt	0.000000	1.225000	0.000000
	vt	0.000000	1.000000	0.000000
	vt	2.000000	1.900000	0.000000

	vt	2.000000	1.675000	0.000000
	vt	2.000000	1.450000	0.000000
690	vt	2.000000	1.225000	0.000000
	vt	2.000000	1.000000	0.000000
	vt	1.750000	1.900000	0.000000
	vt	1.750000	1.675000	0.000000
	vt	1.750000	1.450000	0.000000
695	vt	1.750000	1.225000	0.000000
	vt	1.750000	1.000000	0.000000
	vt	1.500000	1.900000	0.000000
	vt	1.500000	1.675000	0.000000
	vt	1.500000	1.450000	0.000000
700	vt	1.500000	1.225000	0.000000
	vt	1.500000	1.000000	0.000000
	vt	1.250000	1.900000	0.000000
	vt	1.250000	1.675000	0.000000
	vt	1.250000	1.450000	0.000000
705	vt	1.250000	1.225000	0.000000
	vt	1.250000	1.000000	0.000000
	vt	1.000000	1.900000	0.000000
	vt	1.000000	1.675000	0.000000
	vt	1.000000	1.450000	0.000000
710	vt	1.000000	1.225000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	1.000000	1.900000	0.000000
	vt	1.000000	1.675000	0.000000
	vt	1.000000	1.450000	0.000000
715	vt	1.000000	1.225000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	0.750000	1.900000	0.000000
	vt	0.750000	1.675000	0.000000
	vt	0.750000	1.450000	0.000000
720	vt	0.750000	1.225000	0.000000
	vt	0.750000	1.000000	0.000000
	vt	0.500000	1.900000	0.000000
	vt	0.500000	1.675000	0.000000
	vt	0.500000	1.450000	0.000000
725	vt	0.500000	1.225000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.250000	1.900000	0.000000
	vt	0.250000	1.675000	0.000000
	vt	0.250000	1.450000	0.000000
730	vt	0.250000	1.225000	0.000000
	vt	0.250000	1.000000	0.000000
	vt	0.000000	1.900000	0.000000
	vt	0.000000	1.675000	0.000000
	vt	0.000000	1.450000	0.000000
735	vt	0.000000	1.225000	0.000000
	vt	0.000000	1.000000	0.000000
	vt	2.000000	1.000000	0.000000
	vt	2.000000	0.850000	0.000000
	vt	2.000000	0.700000	0.000000
740	vt	2.000000	0.550000	0.000000
	vt	2.000000	0.400000	0.000000
	vt	1.750000	1.000000	0.000000
	vt	1.750000	0.850000	0.000000
	vt	1.750000	0.700000	0.000000

745	vt	1.750000	0.550000	0.000000
	vt	1.750000	0.400000	0.000000
	vt	1.500000	1.000000	0.000000
	vt	1.500000	0.850000	0.000000
	vt	1.500000	0.700000	0.000000
750	vt	1.500000	0.550000	0.000000
	vt	1.500000	0.400000	0.000000
	vt	1.250000	1.000000	0.000000
	vt	1.250000	0.850000	0.000000
	vt	1.250000	0.700000	0.000000
755	vt	1.250000	0.550000	0.000000
	vt	1.250000	0.400000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	1.000000	0.850000	0.000000
	vt	1.000000	0.700000	0.000000
760	vt	1.000000	0.550000	0.000000
	vt	1.000000	0.400000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	1.000000	0.850000	0.000000
	vt	1.000000	0.700000	0.000000
765	vt	1.000000	0.550000	0.000000
	vt	1.000000	0.400000	0.000000
	vt	0.750000	1.000000	0.000000
	vt	0.750000	0.850000	0.000000
	vt	0.750000	0.700000	0.000000
770	vt	0.750000	0.550000	0.000000
	vt	0.750000	0.400000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.500000	0.850000	0.000000
	vt	0.500000	0.700000	0.000000
775	vt	0.500000	0.550000	0.000000
	vt	0.500000	0.400000	0.000000
	vt	0.250000	1.000000	0.000000
	vt	0.250000	0.850000	0.000000
	vt	0.250000	0.700000	0.000000
780	vt	0.250000	0.550000	0.000000
	vt	0.250000	0.400000	0.000000
	vt	0.000000	1.000000	0.000000
	vt	0.000000	0.850000	0.000000
	vt	0.000000	0.700000	0.000000
785	vt	0.000000	0.550000	0.000000
	vt	0.000000	0.400000	0.000000
	vt	2.000000	1.000000	0.000000
	vt	2.000000	0.850000	0.000000
	vt	2.000000	0.700000	0.000000
790	vt	2.000000	0.550000	0.000000
	vt	2.000000	0.400000	0.000000
	vt	1.750000	1.000000	0.000000
	vt	1.750000	0.850000	0.000000
	vt	1.750000	0.700000	0.000000
795	vt	1.750000	0.550000	0.000000
	vt	1.750000	0.400000	0.000000
	vt	1.500000	1.000000	0.000000
	vt	1.500000	0.850000	0.000000
	vt	1.500000	0.700000	0.000000
800	vt	1.500000	0.550000	0.000000
	vt	1.500000	0.400000	0.000000

	vt	1.250000	1.000000	0.000000
	vt	1.250000	0.850000	0.000000
	vt	1.250000	0.700000	0.000000
805	vt	1.250000	0.550000	0.000000
	vt	1.250000	0.400000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	1.000000	0.850000	0.000000
	vt	1.000000	0.700000	0.000000
810	vt	1.000000	0.550000	0.000000
	vt	1.000000	0.400000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	1.000000	0.850000	0.000000
	vt	1.000000	0.700000	0.000000
815	vt	1.000000	0.550000	0.000000
	vt	1.000000	0.400000	0.000000
	vt	0.750000	1.000000	0.000000
	vt	0.750000	0.850000	0.000000
	vt	0.750000	0.700000	0.000000
820	vt	0.750000	0.550000	0.000000
	vt	0.750000	0.400000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.500000	0.850000	0.000000
	vt	0.500000	0.700000	0.000000
825	vt	0.500000	0.550000	0.000000
	vt	0.500000	0.400000	0.000000
	vt	0.250000	1.000000	0.000000
	vt	0.250000	0.850000	0.000000
	vt	0.250000	0.700000	0.000000
830	vt	0.250000	0.550000	0.000000
	vt	0.250000	0.400000	0.000000
	vt	0.000000	1.000000	0.000000
	vt	0.000000	0.850000	0.000000
	vt	0.000000	0.700000	0.000000
835	vt	0.000000	0.550000	0.000000
	vt	0.000000	0.400000	0.000000
	vt	2.000000	0.400000	0.000000
	vt	2.000000	0.300000	0.000000
	vt	2.000000	0.200000	0.000000
840	vt	2.000000	0.100000	0.000000
	vt	2.000000	0.000000	0.000000
	vt	1.750000	0.400000	0.000000
	vt	1.750000	0.300000	0.000000
	vt	1.750000	0.200000	0.000000
845	vt	1.750000	0.100000	0.000000
	vt	1.750000	0.000000	0.000000
	vt	1.500000	0.400000	0.000000
	vt	1.500000	0.300000	0.000000
	vt	1.500000	0.200000	0.000000
850	vt	1.500000	0.100000	0.000000
	vt	1.500000	0.000000	0.000000
	vt	1.250000	0.400000	0.000000
	vt	1.250000	0.300000	0.000000
	vt	1.250000	0.200000	0.000000
855	vt	1.250000	0.100000	0.000000
	vt	1.250000	0.000000	0.000000
	vt	1.000000	0.400000	0.000000
	vt	1.000000	0.300000	0.000000

	vt	1.000000	0.200000	0.000000
860	vt	1.000000	0.100000	0.000000
	vt	1.000000	0.000000	0.000000
	vt	1.000000	0.400000	0.000000
	vt	1.000000	0.300000	0.000000
	vt	1.000000	0.200000	0.000000
865	vt	1.000000	0.100000	0.000000
	vt	1.000000	0.000000	0.000000
	vt	0.750000	0.400000	0.000000
	vt	0.750000	0.300000	0.000000
	vt	0.750000	0.200000	0.000000
870	vt	0.750000	0.100000	0.000000
	vt	0.750000	0.000000	0.000000
	vt	0.500000	0.400000	0.000000
	vt	0.500000	0.300000	0.000000
	vt	0.500000	0.200000	0.000000
875	vt	0.500000	0.100000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.250000	0.400000	0.000000
	vt	0.250000	0.300000	0.000000
	vt	0.250000	0.200000	0.000000
880	vt	0.250000	0.100000	0.000000
	vt	0.250000	0.000000	0.000000
	vt	0.000000	0.400000	0.000000
	vt	0.000000	0.300000	0.000000
	vt	0.000000	0.200000	0.000000
885	vt	0.000000	0.100000	0.000000
	vt	0.000000	0.000000	0.000000
	vt	2.000000	0.400000	0.000000
	vt	2.000000	0.300000	0.000000
	vt	2.000000	0.200000	0.000000
890	vt	2.000000	0.100000	0.000000
	vt	2.000000	0.000000	0.000000
	vt	1.750000	0.400000	0.000000
	vt	1.750000	0.300000	0.000000
	vt	1.750000	0.200000	0.000000
895	vt	1.750000	0.100000	0.000000
	vt	1.750000	0.000000	0.000000
	vt	1.500000	0.400000	0.000000
	vt	1.500000	0.300000	0.000000
	vt	1.500000	0.200000	0.000000
900	vt	1.500000	0.100000	0.000000
	vt	1.500000	0.000000	0.000000
	vt	1.250000	0.400000	0.000000
	vt	1.250000	0.300000	0.000000
	vt	1.250000	0.200000	0.000000
905	vt	1.250000	0.100000	0.000000
	vt	1.250000	0.000000	0.000000
	vt	1.000000	0.400000	0.000000
	vt	1.000000	0.300000	0.000000
	vt	1.000000	0.200000	0.000000
910	vt	1.000000	0.100000	0.000000
	vt	1.000000	0.000000	0.000000
	vt	1.000000	0.400000	0.000000
	vt	1.000000	0.300000	0.000000
	vt	1.000000	0.200000	0.000000
915	vt	1.000000	0.100000	0.000000

	vt	1.000000	0.000000	0.000000
	vt	0.750000	0.400000	0.000000
	vt	0.750000	0.300000	0.000000
	vt	0.750000	0.200000	0.000000
920	vt	0.750000	0.100000	0.000000
	vt	0.750000	0.000000	0.000000
	vt	0.500000	0.400000	0.000000
	vt	0.500000	0.300000	0.000000
	vt	0.500000	0.200000	0.000000
925	vt	0.500000	0.100000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.250000	0.400000	0.000000
	vt	0.250000	0.300000	0.000000
	vt	0.250000	0.200000	0.000000
930	vt	0.250000	0.100000	0.000000
	vt	0.250000	0.000000	0.000000
	vt	0.000000	0.400000	0.000000
	vt	0.000000	0.300000	0.000000
	vt	0.000000	0.200000	0.000000
935	vt	0.000000	0.100000	0.000000
	vt	0.000000	0.000000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	1.000000	0.875000	0.000000
	vt	1.000000	0.750000	0.000000
940	vt	1.000000	0.625000	0.000000
	vt	1.000000	0.500000	0.000000
	vt	0.875000	1.000000	0.000000
	vt	0.875000	0.875000	0.000000
	vt	0.875000	0.750000	0.000000
945	vt	0.875000	0.625000	0.000000
	vt	0.875000	0.500000	0.000000
	vt	0.750000	1.000000	0.000000
	vt	0.750000	0.875000	0.000000
	vt	0.750000	0.750000	0.000000
950	vt	0.750000	0.625000	0.000000
	vt	0.750000	0.500000	0.000000
	vt	0.625000	1.000000	0.000000
	vt	0.625000	0.875000	0.000000
	vt	0.625000	0.750000	0.000000
955	vt	0.625000	0.625000	0.000000
	vt	0.625000	0.500000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.500000	0.875000	0.000000
	vt	0.500000	0.750000	0.000000
960	vt	0.500000	0.625000	0.000000
	vt	0.500000	0.500000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.500000	0.875000	0.000000
	vt	0.500000	0.750000	0.000000
965	vt	0.500000	0.625000	0.000000
	vt	0.500000	0.500000	0.000000
	vt	0.375000	1.000000	0.000000
	vt	0.375000	0.875000	0.000000
	vt	0.375000	0.750000	0.000000
970	vt	0.375000	0.625000	0.000000
	vt	0.375000	0.500000	0.000000
	vt	0.250000	1.000000	0.000000

	vt	0.250000	0.875000	0.000000
	vt	0.250000	0.750000	0.000000
975	vt	0.250000	0.625000	0.000000
	vt	0.250000	0.500000	0.000000
	vt	0.125000	1.000000	0.000000
	vt	0.125000	0.875000	0.000000
	vt	0.125000	0.750000	0.000000
980	vt	0.125000	0.625000	0.000000
	vt	0.125000	0.500000	0.000000
	vt	0.000000	1.000000	0.000000
	vt	0.000000	0.875000	0.000000
	vt	0.000000	0.750000	0.000000
985	vt	0.000000	0.625000	0.000000
	vt	0.000000	0.500000	0.000000
	vt	1.000000	0.500000	0.000000
	vt	1.000000	0.375000	0.000000
	vt	1.000000	0.250000	0.000000
990	vt	1.000000	0.125000	0.000000
	vt	1.000000	0.000000	0.000000
	vt	0.875000	0.500000	0.000000
	vt	0.875000	0.375000	0.000000
	vt	0.875000	0.250000	0.000000
995	vt	0.875000	0.125000	0.000000
	vt	0.875000	0.000000	0.000000
	vt	0.750000	0.500000	0.000000
	vt	0.750000	0.375000	0.000000
	vt	0.750000	0.250000	0.000000
1000	vt	0.750000	0.125000	0.000000
	vt	0.750000	0.000000	0.000000
	vt	0.625000	0.500000	0.000000
	vt	0.625000	0.375000	0.000000
	vt	0.625000	0.250000	0.000000
1005	vt	0.625000	0.125000	0.000000
	vt	0.625000	0.000000	0.000000
	vt	0.500000	0.500000	0.000000
	vt	0.500000	0.375000	0.000000
	vt	0.500000	0.250000	0.000000
1010	vt	0.500000	0.125000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.500000	0.500000	0.000000
	vt	0.500000	0.375000	0.000000
	vt	0.500000	0.250000	0.000000
1015	vt	0.500000	0.125000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.375000	0.500000	0.000000
	vt	0.375000	0.375000	0.000000
	vt	0.375000	0.250000	0.000000
1020	vt	0.375000	0.125000	0.000000
	vt	0.375000	0.000000	0.000000
	vt	0.250000	0.500000	0.000000
	vt	0.250000	0.375000	0.000000
	vt	0.250000	0.250000	0.000000
1025	vt	0.250000	0.125000	0.000000
	vt	0.250000	0.000000	0.000000
	vt	0.125000	0.500000	0.000000
	vt	0.125000	0.375000	0.000000
	vt	0.125000	0.250000	0.000000

1030	vt	0.125000	0.125000	0.000000
	vt	0.125000	0.000000	0.000000
	vt	0.000000	0.500000	0.000000
	vt	0.000000	0.375000	0.000000
	vt	0.000000	0.250000	0.000000
1035	vt	0.000000	0.125000	0.000000
	vt	0.000000	0.000000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.500000	0.225000	0.000000
	vt	0.500000	0.450000	0.000000
1040	vt	0.500000	0.675000	0.000000
	vt	0.500000	0.900000	0.000000
	vt	0.625000	0.000000	0.000000
	vt	0.625000	0.225000	0.000000
	vt	0.625000	0.450000	0.000000
1045	vt	0.625000	0.675000	0.000000
	vt	0.625000	0.900000	0.000000
	vt	0.750000	0.000000	0.000000
	vt	0.750000	0.225000	0.000000
	vt	0.750000	0.450000	0.000000
1050	vt	0.750000	0.675000	0.000000
	vt	0.750000	0.900000	0.000000
	vt	0.875000	0.000000	0.000000
	vt	0.875000	0.225000	0.000000
	vt	0.875000	0.450000	0.000000
1055	vt	0.875000	0.675000	0.000000
	vt	0.875000	0.900000	0.000000
	vt	1.000000	0.000000	0.000000
	vt	1.000000	0.225000	0.000000
	vt	1.000000	0.450000	0.000000
1060	vt	1.000000	0.675000	0.000000
	vt	1.000000	0.900000	0.000000
	vt	0.000000	0.000000	0.000000
	vt	0.000000	0.225000	0.000000
	vt	0.000000	0.450000	0.000000
1065	vt	0.000000	0.675000	0.000000
	vt	0.000000	0.900000	0.000000
	vt	0.125000	0.000000	0.000000
	vt	0.125000	0.225000	0.000000
	vt	0.125000	0.450000	0.000000
1070	vt	0.125000	0.675000	0.000000
	vt	0.125000	0.900000	0.000000
	vt	0.250000	0.000000	0.000000
	vt	0.250000	0.225000	0.000000
	vt	0.250000	0.450000	0.000000
1075	vt	0.250000	0.675000	0.000000
	vt	0.250000	0.900000	0.000000
	vt	0.375000	0.000000	0.000000
	vt	0.375000	0.225000	0.000000
	vt	0.375000	0.450000	0.000000
1080	vt	0.375000	0.675000	0.000000
	vt	0.375000	0.900000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.500000	0.225000	0.000000
	vt	0.500000	0.450000	0.000000
1085	vt	0.500000	0.675000	0.000000
	vt	0.500000	0.900000	0.000000

	vt	0.500000	0.900000	0.000000
	vt	0.500000	0.925000	0.000000
	vt	0.500000	0.950000	0.000000
1090	vt	0.500000	0.975000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.625000	0.900000	0.000000
	vt	0.625000	0.925000	0.000000
	vt	0.625000	0.950000	0.000000
1095	vt	0.625000	0.975000	0.000000
	vt	0.625000	1.000000	0.000000
	vt	0.750000	0.900000	0.000000
	vt	0.750000	0.925000	0.000000
	vt	0.750000	0.950000	0.000000
1100	vt	0.750000	0.975000	0.000000
	vt	0.750000	1.000000	0.000000
	vt	0.875000	0.900000	0.000000
	vt	0.875000	0.925000	0.000000
	vt	0.875000	0.950000	0.000000
1105	vt	0.875000	0.975000	0.000000
	vt	0.875000	1.000000	0.000000
	vt	1.000000	0.900000	0.000000
	vt	1.000000	0.925000	0.000000
	vt	1.000000	0.950000	0.000000
1110	vt	1.000000	0.975000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	0.000000	0.900000	0.000000
	vt	0.000000	0.925000	0.000000
	vt	0.000000	0.950000	0.000000
1115	vt	0.000000	0.975000	0.000000
	vt	0.000000	1.000000	0.000000
	vt	0.125000	0.900000	0.000000
	vt	0.125000	0.925000	0.000000
	vt	0.125000	0.950000	0.000000
1120	vt	0.125000	0.975000	0.000000
	vt	0.125000	1.000000	0.000000
	vt	0.250000	0.900000	0.000000
	vt	0.250000	0.925000	0.000000
	vt	0.250000	0.950000	0.000000
1125	vt	0.250000	0.975000	0.000000
	vt	0.250000	1.000000	0.000000
	vt	0.375000	0.900000	0.000000
	vt	0.375000	0.925000	0.000000
	vt	0.375000	0.950000	0.000000
1130	vt	0.375000	0.975000	0.000000
	vt	0.375000	1.000000	0.000000
	vt	0.500000	0.900000	0.000000
	vt	0.500000	0.925000	0.000000
	vt	0.500000	0.950000	0.000000
1135	vt	0.500000	0.975000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	1.000000	0.750000	0.000000
	vt	1.000000	0.500000	0.000000
1140	vt	1.000000	0.250000	0.000000
	vt	1.000000	0.000000	0.000000
	vt	0.875000	1.000000	0.000000
	vt	0.875000	0.750000	0.000000

	vt	0.875000	0.500000	0.000000
1145	vt	0.875000	0.250000	0.000000
	vt	0.875000	0.000000	0.000000
	vt	0.750000	1.000000	0.000000
	vt	0.750000	0.750000	0.000000
	vt	0.750000	0.500000	0.000000
1150	vt	0.750000	0.250000	0.000000
	vt	0.750000	0.000000	0.000000
	vt	0.625000	1.000000	0.000000
	vt	0.625000	0.750000	0.000000
	vt	0.625000	0.500000	0.000000
1155	vt	0.625000	0.250000	0.000000
	vt	0.625000	0.000000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.500000	0.750000	0.000000
	vt	0.500000	0.500000	0.000000
1160	vt	0.500000	0.250000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.500000	0.750000	0.000000
	vt	0.500000	0.500000	0.000000
1165	vt	0.500000	0.250000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.375000	1.000000	0.000000
	vt	0.375000	0.750000	0.000000
	vt	0.375000	0.500000	0.000000
1170	vt	0.375000	0.250000	0.000000
	vt	0.375000	0.000000	0.000000
	vt	0.250000	1.000000	0.000000
	vt	0.250000	0.750000	0.000000
	vt	0.250000	0.500000	0.000000
1175	vt	0.250000	0.250000	0.000000
	vt	0.250000	0.000000	0.000000
	vt	0.125000	1.000000	0.000000
	vt	0.125000	0.750000	0.000000
	vt	0.125000	0.500000	0.000000
1180	vt	0.125000	0.250000	0.000000
	vt	0.125000	0.000000	0.000000
	vt	0.000000	1.000000	0.000000
	vt	0.000000	0.750000	0.000000
	vt	0.000000	0.500000	0.000000
1185	vt	0.000000	0.250000	0.000000
	vt	0.000000	0.000000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	1.000000	0.750000	0.000000
	vt	1.000000	0.500000	0.000000
1190	vt	1.000000	0.250000	0.000000
	vt	1.000000	0.000000	0.000000
	vt	0.875000	1.000000	0.000000
	vt	0.875000	0.750000	0.000000
	vt	0.875000	0.500000	0.000000
1195	vt	0.875000	0.250000	0.000000
	vt	0.875000	0.000000	0.000000
	vt	0.750000	1.000000	0.000000
	vt	0.750000	0.750000	0.000000
	vt	0.750000	0.500000	0.000000
1200	vt	0.750000	0.250000	0.000000

	vt	0.750000	0.000000	0.000000
	vt	0.625000	1.000000	0.000000
	vt	0.625000	0.750000	0.000000
	vt	0.625000	0.500000	0.000000
1205	vt	0.625000	0.250000	0.000000
	vt	0.625000	0.000000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.500000	0.750000	0.000000
	vt	0.500000	0.500000	0.000000
1210	vt	0.500000	0.250000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.500000	0.750000	0.000000
	vt	0.500000	0.500000	0.000000
1215	vt	0.500000	0.250000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.375000	1.000000	0.000000
	vt	0.375000	0.750000	0.000000
	vt	0.375000	0.500000	0.000000
1220	vt	0.375000	0.250000	0.000000
	vt	0.375000	0.000000	0.000000
	vt	0.250000	1.000000	0.000000
	vt	0.250000	0.750000	0.000000
	vt	0.250000	0.500000	0.000000
1225	vt	0.250000	0.250000	0.000000
	vt	0.250000	0.000000	0.000000
	vt	0.125000	1.000000	0.000000
	vt	0.125000	0.750000	0.000000
	vt	0.125000	0.500000	0.000000
1230	vt	0.125000	0.250000	0.000000
	vt	0.125000	0.000000	0.000000
	vt	0.000000	1.000000	0.000000
	vt	0.000000	0.750000	0.000000
	vt	0.000000	0.500000	0.000000
1235	vt	0.000000	0.250000	0.000000
	vt	0.000000	0.000000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	1.000000	0.750000	0.000000
	vt	1.000000	0.500000	0.000000
1240	vt	1.000000	0.250000	0.000000
	vt	1.000000	0.000000	0.000000
	vt	0.875000	1.000000	0.000000
	vt	0.875000	0.750000	0.000000
	vt	0.875000	0.500000	0.000000
1245	vt	0.875000	0.250000	0.000000
	vt	0.875000	0.000000	0.000000
	vt	0.750000	1.000000	0.000000
	vt	0.750000	0.750000	0.000000
	vt	0.750000	0.500000	0.000000
1250	vt	0.750000	0.250000	0.000000
	vt	0.750000	0.000000	0.000000
	vt	0.625000	1.000000	0.000000
	vt	0.625000	0.750000	0.000000
	vt	0.625000	0.500000	0.000000
1255	vt	0.625000	0.250000	0.000000
	vt	0.625000	0.000000	0.000000
	vt	0.500000	1.000000	0.000000

	vt	0.500000	0.750000	0.000000
	vt	0.500000	0.500000	0.000000
1260	vt	0.500000	0.250000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.500000	0.750000	0.000000
	vt	0.500000	0.500000	0.000000
1265	vt	0.500000	0.250000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.375000	1.000000	0.000000
	vt	0.375000	0.750000	0.000000
	vt	0.375000	0.500000	0.000000
1270	vt	0.375000	0.250000	0.000000
	vt	0.375000	0.000000	0.000000
	vt	0.250000	1.000000	0.000000
	vt	0.250000	0.750000	0.000000
	vt	0.250000	0.500000	0.000000
1275	vt	0.250000	0.250000	0.000000
	vt	0.250000	0.000000	0.000000
	vt	0.125000	1.000000	0.000000
	vt	0.125000	0.750000	0.000000
	vt	0.125000	0.500000	0.000000
1280	vt	0.125000	0.250000	0.000000
	vt	0.125000	0.000000	0.000000
	vt	0.000000	1.000000	0.000000
	vt	0.000000	0.750000	0.000000
	vt	0.000000	0.500000	0.000000
1285	vt	0.000000	0.250000	0.000000
	vt	0.000000	0.000000	0.000000
	vt	1.000000	1.000000	0.000000
	vt	1.000000	0.750000	0.000000
	vt	1.000000	0.500000	0.000000
1290	vt	1.000000	0.250000	0.000000
	vt	1.000000	0.000000	0.000000
	vt	0.875000	1.000000	0.000000
	vt	0.875000	0.750000	0.000000
	vt	0.875000	0.500000	0.000000
1295	vt	0.875000	0.250000	0.000000
	vt	0.875000	0.000000	0.000000
	vt	0.750000	1.000000	0.000000
	vt	0.750000	0.750000	0.000000
	vt	0.750000	0.500000	0.000000
1300	vt	0.750000	0.250000	0.000000
	vt	0.750000	0.000000	0.000000
	vt	0.625000	1.000000	0.000000
	vt	0.625000	0.750000	0.000000
	vt	0.625000	0.500000	0.000000
1305	vt	0.625000	0.250000	0.000000
	vt	0.625000	0.000000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.500000	0.750000	0.000000
	vt	0.500000	0.500000	0.000000
1310	vt	0.500000	0.250000	0.000000
	vt	0.500000	0.000000	0.000000
	vt	0.500000	1.000000	0.000000
	vt	0.500000	0.750000	0.000000
	vt	0.500000	0.500000	0.000000

```
1315 vt 0.500000 0.250000 0.000000
      vt 0.500000 0.000000 0.000000
      vt 0.375000 1.000000 0.000000
      vt 0.375000 0.750000 0.000000
      vt 0.375000 0.500000 0.000000
1320 vt 0.375000 0.250000 0.000000
      vt 0.375000 0.000000 0.000000
      vt 0.250000 1.000000 0.000000
      vt 0.250000 0.750000 0.000000
      vt 0.250000 0.500000 0.000000
1325 vt 0.250000 0.250000 0.000000
      vt 0.250000 0.000000 0.000000
      vt 0.125000 1.000000 0.000000
      vt 0.125000 0.750000 0.000000
      vt 0.125000 0.500000 0.000000
1330 vt 0.125000 0.250000 0.000000
      vt 0.125000 0.000000 0.000000
      vt 0.000000 1.000000 0.000000
      vt 0.000000 0.750000 0.000000
      vt 0.000000 0.500000 0.000000
1335 vt 0.000000 0.250000 0.000000
      vt 0.000000 0.000000 0.000000
      # 800 texture vertices

      vn -0.966742 -0.255752 0.000000
1340 vn -0.966824 0.255443 0.000000
      vn -0.092052 0.995754 0.000000
      vn 0.682050 0.731305 0.000000
      vn 0.870301 0.492521 0.000000
      vn -0.893014 -0.256345 -0.369882
1345 vn -0.893437 0.255996 -0.369102
      vn -0.083877 0.995843 -0.035507
      vn 0.629724 0.731860 0.260439
      vn 0.803725 0.493370 0.332584
      vn -0.683407 -0.256728 -0.683407
1350 vn -0.683531 0.256068 -0.683531
      vn -0.064925 0.995776 -0.064925
      vn 0.481399 0.732469 0.481399
      vn 0.614804 0.493997 0.614804
      vn -0.369882 -0.256345 -0.893014
1355 vn -0.369102 0.255996 -0.893437
      vn -0.035507 0.995843 -0.083877
      vn 0.260439 0.731860 0.629724
      vn 0.332584 0.493369 0.803725
      vn -0.002848 -0.257863 -0.966177
1360 vn -0.001923 0.254736 -0.967009
      vn -0.000266 0.995734 -0.092270
      vn 0.000024 0.731295 0.682061
      vn -0.000000 0.492521 0.870301
      vn 0.379058 -0.359300 -0.852771
1365 vn 0.377110 0.149085 -0.914091
      vn 0.027502 0.992081 -0.122552
      vn -0.261010 0.726762 0.635367
      vn -0.332485 0.492546 0.804271
      vn 0.663548 -0.410790 -0.625264
1370 vn 0.712664 0.073722 -0.697621
      vn 0.099726 0.987509 -0.121983
```

	vn	-0.487320	0.723754	0.488569
	vn	-0.615242	0.492602	0.615484
	vn	0.880028	-0.332906	-0.338709
1375	vn	0.917276	0.167113	-0.361493
	vn	0.113584	0.992365	-0.048070
	vn	-0.634150	0.727508	0.261889
	vn	-0.804126	0.492634	0.332705
	vn	0.966690	-0.255738	0.010454
1380	vn	0.967442	0.252962	0.008103
	vn	0.093436	0.995624	0.001281
	vn	-0.682167	0.731196	-0.000343
	vn	-0.870322	0.492483	-0.000054
	vn	0.893014	-0.256345	0.369882
1385	vn	0.893437	0.255996	0.369102
	vn	0.083877	0.995843	0.035507
	vn	-0.629724	0.731860	-0.260439
	vn	-0.803725	0.493370	-0.332584
	vn	0.683407	-0.256728	0.683407
1390	vn	0.683531	0.256068	0.683531
	vn	0.064925	0.995776	0.064925
	vn	-0.481399	0.732469	-0.481399
	vn	-0.614804	0.493997	-0.614804
	vn	0.369882	-0.256345	0.893014
1395	vn	0.369102	0.255996	0.893437
	vn	0.035507	0.995843	0.083877
	vn	-0.260439	0.731860	-0.629724
	vn	-0.332584	0.493369	-0.803725
	vn	0.000000	-0.255752	0.966742
1400	vn	0.000000	0.255443	0.966824
	vn	0.000000	0.995754	0.092052
	vn	0.000000	0.731305	-0.682050
	vn	0.000000	0.492521	-0.870301
	vn	-0.369882	-0.256345	0.893014
1405	vn	-0.369102	0.255996	0.893437
	vn	-0.035507	0.995843	0.083877
	vn	0.260439	0.731860	-0.629724
	vn	0.332584	0.493370	-0.803725
	vn	-0.683407	-0.256728	0.683407
1410	vn	-0.683531	0.256068	0.683531
	vn	-0.064925	0.995776	0.064925
	vn	0.481399	0.732469	-0.481399
	vn	0.614804	0.493997	-0.614804
	vn	-0.893014	-0.256345	0.369882
1415	vn	-0.893437	0.255996	0.369102
	vn	-0.083877	0.995843	0.035507
	vn	0.629724	0.731860	-0.260439
	vn	0.803725	0.493369	-0.332584
	vn	0.915321	0.402725	-0.000000
1420	vn	0.941808	0.336151	0.000000
	vn	0.978690	0.205342	0.000000
	vn	0.997804	-0.066240	-0.000000
	vn	0.845438	0.403546	0.349835
	vn	0.869996	0.336859	0.360047
1425	vn	0.904193	0.205791	0.374280
	vn	0.921879	-0.066370	0.381752
	vn	0.646802	0.404096	0.646802
	vn	0.665655	0.337351	0.665655

	vn	0.691923	0.206120	0.691923
1430	vn	0.705543	-0.066480	0.705542
	vn	0.349835	0.403546	0.845438
	vn	0.360047	0.336859	0.869996
	vn	0.374280	0.205791	0.904193
	vn	0.381752	-0.066369	0.921879
1435	vn	0.000000	0.402725	0.915321
	vn	-0.000000	0.336151	0.941808
	vn	0.000000	0.205342	0.978690
	vn	0.000000	-0.066240	0.997804
	vn	-0.349835	0.403546	0.845438
1440	vn	-0.360047	0.336859	0.869996
	vn	-0.374280	0.205791	0.904193
	vn	-0.381752	-0.066370	0.921879
	vn	-0.646802	0.404096	0.646802
	vn	-0.665655	0.337351	0.665655
1445	vn	-0.691923	0.206120	0.691923
	vn	-0.705542	-0.066480	0.705543
	vn	-0.845438	0.403546	0.349835
	vn	-0.869996	0.336859	0.360047
	vn	-0.904193	0.205791	0.374280
1450	vn	-0.921879	-0.066369	0.381752
	vn	-0.915321	0.402725	0.000000
	vn	-0.941808	0.336151	-0.000000
	vn	-0.978690	0.205342	0.000000
	vn	-0.997804	-0.066240	0.000000
1455	vn	-0.845438	0.403546	-0.349835
	vn	-0.869996	0.336859	-0.360047
	vn	-0.904193	0.205791	-0.374280
	vn	-0.921879	-0.066370	-0.381752
	vn	-0.646802	0.404096	-0.646802
1460	vn	-0.665655	0.337351	-0.665655
	vn	-0.691923	0.206120	-0.691923
	vn	-0.705543	-0.066480	-0.705542
	vn	-0.349835	0.403546	-0.845438
	vn	-0.360047	0.336859	-0.869996
1465	vn	-0.374280	0.205791	-0.904193
	vn	-0.381752	-0.066369	-0.921879
	vn	-0.000000	0.402725	-0.915321
	vn	0.000000	0.336151	-0.941808
	vn	-0.000000	0.205342	-0.978690
1470	vn	-0.000000	-0.066240	-0.997804
	vn	0.349835	0.403546	-0.845438
	vn	0.360047	0.336859	-0.869996
	vn	0.374280	0.205791	-0.904193
	vn	0.381752	-0.066370	-0.921879
1475	vn	0.646802	0.404096	-0.646802
	vn	0.665655	0.337351	-0.665655
	vn	0.691923	0.206120	-0.691923
	vn	0.705542	-0.066480	-0.705543
	vn	0.845438	0.403546	-0.349835
1480	vn	0.869996	0.336859	-0.360047
	vn	0.904193	0.205791	-0.374280
	vn	0.921879	-0.066369	-0.381752
	vn	0.900182	-0.435513	0.000000
	vn	0.729611	-0.683863	0.000000
1485	vn	0.693951	-0.720022	-0.000000

	vn	0.793950	-0.607983	0.000000
	vn	0.831437	-0.436180	0.344179
	vn	0.673512	-0.684666	0.278594
	vn	0.640399	-0.720924	0.264874
1490	vn	0.732949	-0.608995	0.303167
	vn	0.636092	-0.436778	0.636092
	vn	0.514965	-0.685290	0.514965
	vn	0.489651	-0.721446	0.489651
	vn	0.560555	-0.609554	0.560555
1495	vn	0.344179	-0.436180	0.831437
	vn	0.278595	-0.684666	0.673512
	vn	0.264874	-0.720924	0.640399
	vn	0.303167	-0.608995	0.732949
	vn	-0.000000	-0.435513	0.900182
1500	vn	-0.000000	-0.683863	0.729611
	vn	0.000000	-0.720022	0.693951
	vn	-0.000000	-0.607983	0.793950
	vn	-0.344179	-0.436180	0.831437
	vn	-0.278594	-0.684666	0.673512
1505	vn	-0.264874	-0.720924	0.640399
	vn	-0.303167	-0.608995	0.732949
	vn	-0.636092	-0.436778	0.636092
	vn	-0.514965	-0.685290	0.514965
	vn	-0.489651	-0.721446	0.489651
1510	vn	-0.560555	-0.609554	0.560555
	vn	-0.831437	-0.436180	0.344179
	vn	-0.673512	-0.684666	0.278595
	vn	-0.640399	-0.720924	0.264874
	vn	-0.732949	-0.608995	0.303167
1515	vn	-0.900182	-0.435513	-0.000000
	vn	-0.729611	-0.683863	-0.000000
	vn	-0.693951	-0.720022	0.000000
	vn	-0.793950	-0.607983	-0.000000
	vn	-0.831437	-0.436180	-0.344179
1520	vn	-0.673512	-0.684666	-0.278594
	vn	-0.640399	-0.720924	-0.264874
	vn	-0.732949	-0.608995	-0.303167
	vn	-0.636092	-0.436778	-0.636092
	vn	-0.514965	-0.685290	-0.514965
1525	vn	-0.489651	-0.721446	-0.489651
	vn	-0.560555	-0.609554	-0.560555
	vn	-0.344179	-0.436180	-0.831437
	vn	-0.278595	-0.684666	-0.673512
	vn	-0.264874	-0.720924	-0.640399
1530	vn	-0.303167	-0.608995	-0.732949
	vn	0.000000	-0.435513	-0.900182
	vn	0.000000	-0.683863	-0.729611
	vn	-0.000000	-0.720022	-0.693951
	vn	0.000000	-0.607983	-0.793950
1535	vn	0.344179	-0.436180	-0.831437
	vn	0.278594	-0.684666	-0.673512
	vn	0.264874	-0.720924	-0.640399
	vn	0.303167	-0.608995	-0.732949
	vn	0.636092	-0.436778	-0.636092
1540	vn	0.514965	-0.685290	-0.514965
	vn	0.489651	-0.721446	-0.489651
	vn	0.560555	-0.609554	-0.560555

	vn	0.831437	-0.436180	-0.344179
	vn	0.673512	-0.684666	-0.278595
1545	vn	0.640399	-0.720924	-0.264874
	vn	0.732949	-0.608995	-0.303167
	vn	0.623860	-0.781536	0.000000
	vn	0.177291	-0.984159	-0.000000
	vn	0.049207	-0.998789	0.000000
1550	vn	0.000000	-1.000000	0.000000
	vn	0.576230	-0.781801	0.238217
	vn	0.163628	-0.984208	0.067527
	vn	0.045422	-0.998792	0.018736
	vn	0.440416	-0.782348	0.440416
1555	vn	0.124903	-0.984276	0.124903
	vn	0.034662	-0.998798	0.034662
	vn	0.238217	-0.781801	0.576230
	vn	0.067527	-0.984208	0.163628
	vn	0.018736	-0.998792	0.045422
1560	vn	-0.000000	-0.781536	0.623860
	vn	0.000000	-0.984159	0.177291
	vn	-0.000000	-0.998789	0.049207
	vn	-0.238217	-0.781801	0.576230
	vn	-0.067527	-0.984208	0.163628
1565	vn	-0.018736	-0.998792	0.045422
	vn	-0.440416	-0.782348	0.440416
	vn	-0.124903	-0.984276	0.124903
	vn	-0.034662	-0.998798	0.034662
	vn	-0.576230	-0.781801	0.238217
1570	vn	-0.163628	-0.984208	0.067527
	vn	-0.045422	-0.998792	0.018736
	vn	-0.623860	-0.781536	-0.000000
	vn	-0.177291	-0.984159	0.000000
	vn	-0.049207	-0.998789	-0.000000
1575	vn	-0.576230	-0.781801	-0.238217
	vn	-0.163628	-0.984208	-0.067527
	vn	-0.045422	-0.998792	-0.018736
	vn	-0.440416	-0.782348	-0.440416
	vn	-0.124903	-0.984276	-0.124903
1580	vn	-0.034662	-0.998798	-0.034662
	vn	-0.238217	-0.781801	-0.576230
	vn	-0.067527	-0.984208	-0.163628
	vn	-0.018736	-0.998792	-0.045422
	vn	0.000000	-0.781536	-0.623860
1585	vn	-0.000000	-0.984159	-0.177291
	vn	0.000000	-0.998789	-0.049207
	vn	0.238217	-0.781801	-0.576230
	vn	0.067527	-0.984208	-0.163628
	vn	0.018736	-0.998792	-0.045422
1590	vn	0.440416	-0.782348	-0.440416
	vn	0.124903	-0.984276	-0.124903
	vn	0.034662	-0.998798	-0.034662
	vn	0.576230	-0.781801	-0.238217
	vn	0.163628	-0.984208	-0.067527
1595	vn	0.045422	-0.998792	-0.018736
	vn	0.007786	-0.999970	-0.000216
	vn	0.039138	-0.999233	-0.000989
	vn	0.179512	-0.983746	-0.004369
	vn	0.612299	-0.790557	-0.010460

1600	vn	0.986152	-0.165708	-0.006670
	vn	0.007039	-0.812495	0.582926
	vn	0.036127	-0.837257	0.545614
	vn	0.161846	-0.810421	0.563048
	vn	0.482365	-0.595148	0.642746
1605	vn	0.738720	-0.114594	0.664199
	vn	-0.001909	0.162121	0.986769
	vn	0.002762	0.017107	0.999850
	vn	0.010533	0.073398	0.997247
	vn	-0.066041	0.130070	0.989303
1610	vn	-0.094427	0.016594	0.995393
	vn	-0.009203	0.871509	0.490293
	vn	-0.048606	0.840609	0.539457
	vn	-0.223298	0.802880	0.552739
	vn	-0.596365	0.559971	0.575135
1615	vn	-0.803337	0.068236	0.591603
	vn	-0.010561	0.999944	0.000103
	vn	-0.058798	0.998270	0.000710
	vn	-0.280710	0.959787	0.003269
	vn	-0.749723	0.661738	0.004268
1620	vn	-0.997351	0.072714	0.002059
	vn	-0.008792	0.871493	-0.490330
	vn	-0.046494	0.841178	-0.538756
	vn	-0.217909	0.806807	-0.549161
	vn	-0.597291	0.560026	-0.574121
1625	vn	-0.804000	0.062913	-0.591292
	vn	-0.001806	0.161691	-0.986840
	vn	0.002031	0.014555	-0.999892
	vn	0.009215	0.060069	-0.998152
	vn	-0.059334	0.113865	-0.991723
1630	vn	-0.086899	0.012290	-0.996141
	vn	0.006418	-0.812379	-0.583095
	vn	0.033783	-0.837512	-0.545373
	vn	0.157113	-0.811947	-0.562190
	vn	0.484406	-0.589366	-0.646528
1635	vn	0.738870	-0.101320	-0.666187
	vn	0.946512	0.322650	-0.003357
	vn	0.825830	0.563870	-0.007452
	vn	0.650011	0.759893	-0.006937
	vn	0.532429	0.846459	-0.005245
1640	vn	0.725608	0.259351	0.637361
	vn	0.645945	0.461988	0.607719
	vn	0.531615	0.636660	0.558614
	vn	0.424964	0.681717	0.595540
	vn	-0.049562	-0.019755	0.998576
1645	vn	-0.037816	-0.035624	0.998650
	vn	-0.037914	-0.036512	0.998614
	vn	-0.168854	-0.297945	0.939530
	vn	-0.742342	-0.299166	0.599523
	vn	-0.619602	-0.529406	0.579502
1650	vn	-0.483708	-0.685760	0.543837
	vn	-0.445293	-0.794355	0.413177
	vn	-0.926513	-0.376258	0.001996
	vn	-0.753920	-0.656952	0.004317
	vn	-0.566224	-0.824244	0.003461
1655	vn	-0.481804	-0.876277	0.001850
	vn	-0.744675	-0.294425	-0.598977

	vn	-0.621949	-0.528114	-0.578165
	vn	-0.481171	-0.688340	-0.542828
	vn	-0.438055	-0.797035	-0.415744
1660	vn	-0.044337	-0.017056	-0.998871
	vn	-0.026176	-0.028166	-0.999260
	vn	-0.025294	-0.028332	-0.999278
	vn	-0.157482	-0.289392	-0.944167
	vn	0.728244	0.252410	-0.637142
1665	vn	0.647055	0.459725	-0.608254
	vn	0.522994	0.640657	-0.562170
	vn	0.409978	0.682857	-0.604669
	vn	-0.230787	0.972982	-0.006523
	vn	-0.548936	0.835863	-0.001511
1670	vn	-0.875671	0.482806	0.009893
	vn	-0.877554	0.479097	0.019092
	vn	-0.696190	0.717439	0.024497
	vn	-0.152877	0.687211	0.710190
	vn	-0.316721	0.637750	0.702113
1675	vn	-0.601067	0.471452	0.645330
	vn	-0.635889	0.446090	0.629801
	vn	-0.435746	0.601008	0.670011
	vn	0.111113	-0.085070	0.990160
	vn	0.223310	0.006540	0.974726
1680	vn	0.190097	0.154964	0.969458
	vn	0.005271	0.189482	0.981870
	vn	-0.011752	0.246688	0.969024
	vn	0.343906	-0.722796	0.599412
	vn	0.572489	-0.567656	0.591627
1685	vn	0.787436	-0.256459	0.560512
	vn	0.647097	-0.306374	0.698141
	vn	0.427528	-0.499343	0.753576
	vn	0.410926	-0.911668	0.001284
	vn	0.671520	-0.740986	-0.000899
1690	vn	0.922026	-0.387060	-0.007253
	vn	0.846910	-0.531556	-0.013854
	vn	0.535924	-0.844201	-0.010505
	vn	0.341188	-0.722823	-0.600931
	vn	0.578664	-0.561139	-0.591838
1695	vn	0.784869	-0.251020	-0.566542
	vn	0.642681	-0.302257	-0.703990
	vn	0.418589	-0.500042	-0.758117
	vn	0.115806	-0.079139	-0.990114
	vn	0.232811	0.012565	-0.972441
1700	vn	0.206662	0.153601	-0.966280
	vn	0.024500	0.161443	-0.986578
	vn	0.003382	0.211115	-0.977455
	vn	-0.134912	0.687491	-0.713551
	vn	-0.319540	0.633073	-0.705063
1705	vn	-0.603902	0.461442	-0.649903
	vn	-0.631815	0.437169	-0.640072
	vn	-0.424306	0.612706	-0.666750
	vn	-0.425801	0.904753	0.010805
	vn	0.022046	0.999756	0.001623
1710	vn	0.999598	0.025875	0.011556
	vn	0.709587	-0.704552	0.009672
	vn	-0.259858	0.791937	0.552548
	vn	0.009539	0.999720	-0.021674

	vn	0.410157	0.332912	-0.849082
1715	vn	0.541523	-0.548619	-0.637001
	vn	0.046311	0.455223	0.889172
	vn	-0.010688	0.988794	0.148900
	vn	-0.044376	0.682946	-0.729120
	vn	0.122824	0.009233	-0.992385
1720	vn	0.481839	-0.180440	0.857480
	vn	0.455272	0.736752	0.499925
	vn	-0.220542	0.907193	-0.358277
	vn	-0.235919	0.657250	-0.715797
	vn	0.728094	-0.685300	-0.015585
1725	vn	0.888738	0.458112	-0.016679
	vn	-0.260098	0.965582	0.000800
	vn	-0.371611	0.928378	-0.004418
	vn	0.480165	-0.178362	-0.858853
	vn	0.488102	0.716802	-0.497947
1730	vn	-0.222004	0.905399	0.361892
	vn	-0.235405	0.663180	0.710477
	vn	0.058720	0.437702	-0.897200
	vn	0.001326	0.986459	-0.164002
	vn	-0.044190	0.681675	0.730319
1735	vn	0.138801	-0.034188	0.989730
	vn	-0.258890	0.797206	-0.545380
	vn	0.012270	0.999739	0.019287
	vn	0.398632	0.354890	0.845663
	vn	0.537564	-0.581398	0.610738
1740	vn	0.000000	1.000000	0.000000
	vn	0.824540	0.565804	0.000017
	vn	0.917701	-0.397272	0.000034
	vn	0.935269	-0.353939	0.000113
	vn	0.780712	0.624890	0.000075
1745	vn	0.762640	0.565035	0.314825
	vn	0.847982	-0.397998	0.350034
	vn	0.864141	-0.355261	0.356441
	vn	0.720992	0.625625	0.297933
	vn	0.583357	0.565165	0.583338
1750	vn	0.648485	-0.398726	0.648448
	vn	0.660872	-0.355894	0.660748
	vn	0.551862	0.625290	0.551780
	vn	0.314824	0.565051	0.762629
	vn	0.350045	-0.397976	0.847988
1755	vn	0.356474	-0.355199	0.864153
	vn	0.297983	0.625515	0.721067
	vn	-0.000017	0.565804	0.824540
	vn	-0.000034	-0.397272	0.917701
	vn	-0.000113	-0.353939	0.935269
1760	vn	-0.000075	0.624890	0.780712
	vn	-0.314825	0.565035	0.762640
	vn	-0.350034	-0.397998	0.847982
	vn	-0.356441	-0.355261	0.864141
	vn	-0.297933	0.625625	0.720992
1765	vn	-0.583338	0.565165	0.583357
	vn	-0.648448	-0.398726	0.648485
	vn	-0.660748	-0.355894	0.660872
	vn	-0.551780	0.625290	0.551862
	vn	-0.762629	0.565051	0.314824
1770	vn	-0.847988	-0.397976	0.350045

	vn	-0.864153	-0.355199	0.356474
	vn	-0.721067	0.625515	0.297983
	vn	-0.824540	0.565804	-0.000017
	vn	-0.917701	-0.397272	-0.000034
1775	vn	-0.935269	-0.353939	-0.000113
	vn	-0.780712	0.624890	-0.000075
	vn	-0.762640	0.565035	-0.314825
	vn	-0.847982	-0.397998	-0.350034
	vn	-0.864141	-0.355261	-0.356441
1780	vn	-0.720992	0.625625	-0.297933
	vn	-0.583357	0.565165	-0.583338
	vn	-0.648485	-0.398726	-0.648448
	vn	-0.660872	-0.355894	-0.660748
	vn	-0.551862	0.625290	-0.551780
1785	vn	-0.314824	0.565051	-0.762629
	vn	-0.350045	-0.397976	-0.847988
	vn	-0.356474	-0.355199	-0.864153
	vn	-0.297983	0.625515	-0.721067
	vn	0.000017	0.565804	-0.824540
1790	vn	0.000034	-0.397272	-0.917701
	vn	0.000113	-0.353939	-0.935269
	vn	0.000075	0.624890	-0.780712
	vn	0.314825	0.565035	-0.762640
	vn	0.350034	-0.397998	-0.847982
1795	vn	0.356441	-0.355261	-0.864141
	vn	0.297933	0.625625	-0.720992
	vn	0.583338	0.565165	-0.583357
	vn	0.648448	-0.398726	-0.648485
	vn	0.660748	-0.355894	-0.660872
1800	vn	0.551780	0.625290	-0.551862
	vn	0.762629	0.565051	-0.314824
	vn	0.847988	-0.397976	-0.350045
	vn	0.864153	-0.355199	-0.356474
	vn	0.721067	0.625515	-0.297983
1805	vn	0.236583	0.971611	-0.000000
	vn	0.173084	0.984907	-0.000000
	vn	0.379703	0.925108	-0.000000
	vn	0.526673	0.850068	0.000000
	vn	0.217978	0.971775	0.090216
1810	vn	0.159589	0.984977	0.065961
	vn	0.350498	0.925311	0.144740
	vn	0.485590	0.850653	0.201474
	vn	0.166631	0.971838	0.166631
	vn	0.121908	0.985026	0.121908
1815	vn	0.267668	0.925585	0.267668
	vn	0.371315	0.851029	0.371315
	vn	0.090216	0.971775	0.217978
	vn	0.065961	0.984977	0.159589
	vn	0.144740	0.925311	0.350498
1820	vn	0.201475	0.850653	0.485590
	vn	0.000000	0.971611	0.236583
	vn	0.000000	0.984907	0.173084
	vn	0.000000	0.925108	0.379703
	vn	-0.000000	0.850068	0.526673
1825	vn	-0.090216	0.971775	0.217978
	vn	-0.065961	0.984977	0.159589
	vn	-0.144740	0.925311	0.350498

```

vn -0.201474 0.850653 0.485590
vn -0.166631 0.971838 0.166631
1830 vn -0.121908 0.985026 0.121908
vn -0.267668 0.925585 0.267668
vn -0.371315 0.851029 0.371315
vn -0.217978 0.971775 0.090216
vn -0.159589 0.984977 0.065961
1835 vn -0.350498 0.925311 0.144740
vn -0.485590 0.850653 0.201475
vn -0.236583 0.971611 0.000000
vn -0.173084 0.984907 0.000000
vn -0.379703 0.925108 0.000000
1840 vn -0.526673 0.850068 -0.000000
vn -0.217978 0.971775 -0.090216
vn -0.159589 0.984977 -0.065961
vn -0.350498 0.925311 -0.144740
vn -0.485590 0.850653 -0.201474
1845 vn -0.166631 0.971838 -0.166631
vn -0.121908 0.985026 -0.121908
vn -0.267668 0.925585 -0.267668
vn -0.371315 0.851029 -0.371315
vn -0.090216 0.971775 -0.217978
1850 vn -0.065961 0.984977 -0.159589
vn -0.144740 0.925311 -0.350498
vn -0.201475 0.850653 -0.485590
vn -0.000000 0.971611 -0.236583
vn -0.000000 0.984907 -0.173084
1855 vn -0.000000 0.925108 -0.379703
vn 0.000000 0.850068 -0.526673
vn 0.090216 0.971775 -0.217978
vn 0.065961 0.984977 -0.159589
vn 0.144740 0.925311 -0.350498
1860 vn 0.201474 0.850653 -0.485590
vn 0.166631 0.971838 -0.166631
vn 0.121908 0.985026 -0.121908
vn 0.267668 0.925585 -0.267668
vn 0.371315 0.851029 -0.371315
1865 vn 0.217978 0.971775 -0.090216
vn 0.159589 0.984977 -0.065961
vn 0.350498 0.925311 -0.144740
vn 0.485590 0.850653 -0.201475
# 530 vertex normals
1870
g Teapot01
f 1/1/1 6/6/6 7/7/7
f 7/7/7 2/2/2 1/1/1
f 2/2/2 7/7/7 8/8/8
1875 f 8/8/8 3/3/3 2/2/2
f 3/3/3 8/8/8 9/9/9
f 9/9/9 4/4/4 3/3/3
f 4/4/4 9/9/9 10/10/10
f 10/10/10 5/5/5 4/4/4
1880 f 6/6/6 11/11/11 12/12/12
f 12/12/12 7/7/7 6/6/6
f 7/7/7 12/12/12 13/13/13
f 13/13/13 8/8/8 7/7/7
f 8/8/8 13/13/13 14/14/14

```


1885	f	14/14/14	9/9/9	8/8/8
	f	9/9/9	14/14/14	15/15/15
	f	15/15/15	10/10/10	9/9/9
	f	11/11/11	16/16/16	17/17/17
	f	17/17/17	12/12/12	11/11/11
1890	f	12/12/12	17/17/17	18/18/18
	f	18/18/18	13/13/13	12/12/12
	f	13/13/13	18/18/18	19/19/19
	f	19/19/19	14/14/14	13/13/13
	f	14/14/14	19/19/19	20/20/20
1895	f	20/20/20	15/15/15	14/14/14
	f	16/16/16	21/21/21	22/22/22
	f	22/22/22	17/17/17	16/16/16
	f	17/17/17	22/22/22	23/23/23
	f	23/23/23	18/18/18	17/17/17
1900	f	18/18/18	23/23/23	24/24/24
	f	24/24/24	19/19/19	18/18/18
	f	19/19/19	24/24/24	25/25/25
	f	25/25/25	20/20/20	19/19/19
	f	21/26/21	26/31/26	27/32/27
1905	f	27/32/27	22/27/22	21/26/21
	f	22/27/22	27/32/27	28/33/28
	f	28/33/28	23/28/23	22/27/22
	f	23/28/23	28/33/28	29/34/29
	f	29/34/29	24/29/24	23/28/23
1910	f	24/29/24	29/34/29	30/35/30
	f	30/35/30	25/30/25	24/29/24
	f	26/31/26	31/36/31	32/37/32
	f	32/37/32	27/32/27	26/31/26
	f	27/32/27	32/37/32	33/38/33
1915	f	33/38/33	28/33/28	27/32/27
	f	28/33/28	33/38/33	34/39/34
	f	34/39/34	29/34/29	28/33/28
	f	29/34/29	34/39/34	35/40/35
	f	35/40/35	30/35/30	29/34/29
1920	f	31/36/31	36/41/36	37/42/37
	f	37/42/37	32/37/32	31/36/31
	f	32/37/32	37/42/37	38/43/38
	f	38/43/38	33/38/33	32/37/32
	f	33/38/33	38/43/38	39/44/39
1925	f	39/44/39	34/39/34	33/38/33
	f	34/39/34	39/44/39	40/45/40
	f	40/45/40	35/40/35	34/39/34
	f	36/41/36	41/46/41	42/47/42
	f	42/47/42	37/42/37	36/41/36
1930	f	37/42/37	42/47/42	43/48/43
	f	43/48/43	38/43/38	37/42/37
	f	38/43/38	43/48/43	44/49/44
	f	44/49/44	39/44/39	38/43/38
	f	39/44/39	44/49/44	45/50/45
1935	f	45/50/45	40/45/40	39/44/39
	f	41/51/41	46/56/46	47/57/47
	f	47/57/47	42/52/42	41/51/41
	f	42/52/42	47/57/47	48/58/48
	f	48/58/48	43/53/43	42/52/42
1940	f	43/53/43	48/58/48	49/59/49
	f	49/59/49	44/54/44	43/53/43

	f	44/54/44	49/59/49	50/60/50
	f	50/60/50	45/55/45	44/54/44
	f	46/56/46	51/61/51	52/62/52
1945	f	52/62/52	47/57/47	46/56/46
	f	47/57/47	52/62/52	53/63/53
	f	53/63/53	48/58/48	47/57/47
	f	48/58/48	53/63/53	54/64/54
	f	54/64/54	49/59/49	48/58/48
1950	f	49/59/49	54/64/54	55/65/55
	f	55/65/55	50/60/50	49/59/49
	f	51/61/51	56/66/56	57/67/57
	f	57/67/57	52/62/52	51/61/51
	f	52/62/52	57/67/57	58/68/58
1955	f	58/68/58	53/63/53	52/62/52
	f	53/63/53	58/68/58	59/69/59
	f	59/69/59	54/64/54	53/63/53
	f	54/64/54	59/69/59	60/70/60
	f	60/70/60	55/65/55	54/64/54
1960	f	56/66/56	61/71/61	62/72/62
	f	62/72/62	57/67/57	56/66/56
	f	57/67/57	62/72/62	63/73/63
	f	63/73/63	58/68/58	57/67/57
	f	58/68/58	63/73/63	64/74/64
1965	f	64/74/64	59/69/59	58/68/58
	f	59/69/59	64/74/64	65/75/65
	f	65/75/65	60/70/60	59/69/59
	f	61/76/61	66/81/66	67/82/67
	f	67/82/67	62/77/62	61/76/61
1970	f	62/77/62	67/82/67	68/83/68
	f	68/83/68	63/78/63	62/77/62
	f	63/78/63	68/83/68	69/84/69
	f	69/84/69	64/79/64	63/78/63
	f	64/79/64	69/84/69	70/85/70
1975	f	70/85/70	65/80/65	64/79/64
	f	66/81/66	71/86/71	72/87/72
	f	72/87/72	67/82/67	66/81/66
	f	67/82/67	72/87/72	73/88/73
	f	73/88/73	68/83/68	67/82/67
1980	f	68/83/68	73/88/73	74/89/74
	f	74/89/74	69/84/69	68/83/68
	f	69/84/69	74/89/74	75/90/75
	f	75/90/75	70/85/70	69/84/69
	f	71/86/71	76/91/76	77/92/77
1985	f	77/92/77	72/87/72	71/86/71
	f	72/87/72	77/92/77	78/93/78
	f	78/93/78	73/88/73	72/87/72
	f	73/88/73	78/93/78	79/94/79
	f	79/94/79	74/89/74	73/88/73
1990	f	74/89/74	79/94/79	80/95/80
	f	80/95/80	75/90/75	74/89/74
	f	76/91/76	1/96/1	2/97/2
	f	2/97/2	77/92/77	76/91/76
	f	77/92/77	2/97/2	3/98/3
1995	f	3/98/3	78/93/78	77/92/77
	f	78/93/78	3/98/3	4/99/4
	f	4/99/4	79/94/79	78/93/78
	f	79/94/79	4/99/4	5/100/5

```

f 5/100/5 80/95/80 79/94/79
2000 f 5/101/5 10/106/10 85/107/85
f 85/107/85 81/102/81 5/101/5
f 81/102/81 85/107/85 86/108/86
f 86/108/86 82/103/82 81/102/81
f 82/103/82 86/108/86 87/109/87
2005 f 87/109/87 83/104/83 82/103/82
f 83/104/83 87/109/87 88/110/88
f 88/110/88 84/105/84 83/104/83
f 10/106/10 15/111/15 89/112/89
f 89/112/89 85/107/85 10/106/10
2010 f 85/107/85 89/112/89 90/113/90
f 90/113/90 86/108/86 85/107/85
f 86/108/86 90/113/90 91/114/91
f 91/114/91 87/109/87 86/108/86
f 87/109/87 91/114/91 92/115/92
2015 f 92/115/92 88/110/88 87/109/87
f 15/111/15 20/116/20 93/117/93
f 93/117/93 89/112/89 15/111/15
f 89/112/89 93/117/93 94/118/94
f 94/118/94 90/113/90 89/112/89
2020 f 90/113/90 94/118/94 95/119/95
f 95/119/95 91/114/91 90/113/90
f 91/114/91 95/119/95 96/120/96
f 96/120/96 92/115/92 91/114/91
f 20/116/20 25/121/25 97/122/97
2025 f 97/122/97 93/117/93 20/116/20
f 93/117/93 97/122/97 98/123/98
f 98/123/98 94/118/94 93/117/93
f 94/118/94 98/123/98 99/124/99
f 99/124/99 95/119/95 94/118/94
2030 f 95/119/95 99/124/99 100/125/100
f 100/125/100 96/120/96 95/119/95
f 25/126/25 30/131/30 101/132/101
f 101/132/101 97/127/97 25/126/25
f 97/127/97 101/132/101 102/133/102
2035 f 102/133/102 98/128/98 97/127/97
f 98/128/98 102/133/102 103/134/103
f 103/134/103 99/129/99 98/128/98
f 99/129/99 103/134/103 104/135/104
f 104/135/104 100/130/100 99/129/99
2040 f 30/131/30 35/136/35 105/137/105
f 105/137/105 101/132/101 30/131/30
f 101/132/101 105/137/105 106/138/106
f 106/138/106 102/133/102 101/132/101
f 102/133/102 106/138/106 107/139/107
2045 f 107/139/107 103/134/103 102/133/102
f 103/134/103 107/139/107 108/140/108
f 108/140/108 104/135/104 103/134/103
f 35/136/35 40/141/40 109/142/109
f 109/142/109 105/137/105 35/136/35
2050 f 105/137/105 109/142/109 110/143/110
f 110/143/110 106/138/106 105/137/105
f 106/138/106 110/143/110 111/144/111
f 111/144/111 107/139/107 106/138/106
f 107/139/107 111/144/111 112/145/112
2055 f 112/145/112 108/140/108 107/139/107

```

f 40/141/40 45/146/45 113/147/113
 f 113/147/113 109/142/109 40/141/40
 f 109/142/109 113/147/113 114/148/114
 f 114/148/114 110/143/110 109/142/109
 2060 f 110/143/110 114/148/114 115/149/115
 f 115/149/115 111/144/111 110/143/110
 f 111/144/111 115/149/115 116/150/116
 f 116/150/116 112/145/112 111/144/111
 f 45/151/45 50/156/50 117/157/117
 2065 f 117/157/117 113/152/113 45/151/45
 f 113/152/113 117/157/117 118/158/118
 f 118/158/118 114/153/114 113/152/113
 f 114/153/114 118/158/118 119/159/119
 f 119/159/119 115/154/115 114/153/114
 2070 f 115/154/115 119/159/119 120/160/120
 f 120/160/120 116/155/116 115/154/115
 f 50/156/50 55/161/55 121/162/121
 f 121/162/121 117/157/117 50/156/50
 f 117/157/117 121/162/121 122/163/122
 2075 f 122/163/122 118/158/118 117/157/117
 f 118/158/118 122/163/122 123/164/123
 f 123/164/123 119/159/119 118/158/118
 f 119/159/119 123/164/123 124/165/124
 f 124/165/124 120/160/120 119/159/119
 2080 f 55/161/55 60/166/60 125/167/125
 f 125/167/125 121/162/121 55/161/55
 f 121/162/121 125/167/125 126/168/126
 f 126/168/126 122/163/122 121/162/121
 f 122/163/122 126/168/126 127/169/127
 2085 f 127/169/127 123/164/123 122/163/122
 f 123/164/123 127/169/127 128/170/128
 f 128/170/128 124/165/124 123/164/123
 f 60/166/60 65/171/65 129/172/129
 f 129/172/129 125/167/125 60/166/60
 2090 f 125/167/125 129/172/129 130/173/130
 f 130/173/130 126/168/126 125/167/125
 f 126/168/126 130/173/130 131/174/131
 f 131/174/131 127/169/127 126/168/126
 f 127/169/127 131/174/131 132/175/132
 2095 f 132/175/132 128/170/128 127/169/127
 f 65/176/65 70/181/70 133/182/133
 f 133/182/133 129/177/129 65/176/65
 f 129/177/129 133/182/133 134/183/134
 f 134/183/134 130/178/130 129/177/129
 2100 f 130/178/130 134/183/134 135/184/135
 f 135/184/135 131/179/131 130/178/130
 f 131/179/131 135/184/135 136/185/136
 f 136/185/136 132/180/132 131/179/131
 f 70/181/70 75/186/75 137/187/137
 2105 f 137/187/137 133/182/133 70/181/70
 f 133/182/133 137/187/137 138/188/138
 f 138/188/138 134/183/134 133/182/133
 f 134/183/134 138/188/138 139/189/139
 f 139/189/139 135/184/135 134/183/134
 2110 f 135/184/135 139/189/139 140/190/140
 f 140/190/140 136/185/136 135/184/135
 f 75/186/75 80/191/80 141/192/141

f 141/192/141 137/187/137 75/186/75
 f 137/187/137 141/192/141 142/193/142
 2115 f 142/193/142 138/188/138 137/187/137
 f 138/188/138 142/193/142 143/194/143
 f 143/194/143 139/189/139 138/188/138
 f 139/189/139 143/194/143 144/195/144
 f 144/195/144 140/190/140 139/189/139
 2120 f 80/191/80 5/196/5 81/197/81
 f 81/197/81 141/192/141 80/191/80
 f 141/192/141 81/197/81 82/198/82
 f 82/198/82 142/193/142 141/192/141
 f 142/193/142 82/198/82 83/199/83
 2125 f 83/199/83 143/194/143 142/193/142
 f 143/194/143 83/199/83 84/200/84
 f 84/200/84 144/195/144 143/194/143
 f 84/201/84 88/206/88 149/207/149
 f 149/207/149 145/202/145 84/201/84
 2130 f 145/202/145 149/207/149 150/208/150
 f 150/208/150 146/203/146 145/202/145
 f 146/203/146 150/208/150 151/209/151
 f 151/209/151 147/204/147 146/203/146
 f 147/204/147 151/209/151 152/210/152
 2135 f 152/210/152 148/205/148 147/204/147
 f 88/206/88 92/211/92 153/212/153
 f 153/212/153 149/207/149 88/206/88
 f 149/207/149 153/212/153 154/213/154
 f 154/213/154 150/208/150 149/207/149
 2140 f 150/208/150 154/213/154 155/214/155
 f 155/214/155 151/209/151 150/208/150
 f 151/209/151 155/214/155 156/215/156
 f 156/215/156 152/210/152 151/209/151
 f 92/211/92 96/216/96 157/217/157
 2145 f 157/217/157 153/212/153 92/211/92
 f 153/212/153 157/217/157 158/218/158
 f 158/218/158 154/213/154 153/212/153
 f 154/213/154 158/218/158 159/219/159
 f 159/219/159 155/214/155 154/213/154
 2150 f 155/214/155 159/219/159 160/220/160
 f 160/220/160 156/215/156 155/214/155
 f 96/216/96 100/221/100 161/222/161
 f 161/222/161 157/217/157 96/216/96
 f 157/217/157 161/222/161 162/223/162
 2155 f 162/223/162 158/218/158 157/217/157
 f 158/218/158 162/223/162 163/224/163
 f 163/224/163 159/219/159 158/218/158
 f 159/219/159 163/224/163 164/225/164
 f 164/225/164 160/220/160 159/219/159
 2160 f 100/226/100 104/231/104 165/232/165
 f 165/232/165 161/227/161 100/226/100
 f 161/227/161 165/232/165 166/233/166
 f 166/233/166 162/228/162 161/227/161
 f 162/228/162 166/233/166 167/234/167
 2165 f 167/234/167 163/229/163 162/228/162
 f 163/229/163 167/234/167 168/235/168
 f 168/235/168 164/230/164 163/229/163
 f 104/231/104 108/236/108 169/237/169
 f 169/237/169 165/232/165 104/231/104

2170	f	165/232/165	169/237/169	170/238/170
	f	170/238/170	166/233/166	165/232/165
	f	166/233/166	170/238/170	171/239/171
	f	171/239/171	167/234/167	166/233/166
	f	167/234/167	171/239/171	172/240/172
2175	f	172/240/172	168/235/168	167/234/167
	f	108/236/108	112/241/112	173/242/173
	f	173/242/173	169/237/169	108/236/108
	f	169/237/169	173/242/173	174/243/174
	f	174/243/174	170/238/170	169/237/169
2180	f	170/238/170	174/243/174	175/244/175
	f	175/244/175	171/239/171	170/238/170
	f	171/239/171	175/244/175	176/245/176
	f	176/245/176	172/240/172	171/239/171
	f	112/241/112	116/246/116	177/247/177
2185	f	177/247/177	173/242/173	112/241/112
	f	173/242/173	177/247/177	178/248/178
	f	178/248/178	174/243/174	173/242/173
	f	174/243/174	178/248/178	179/249/179
	f	179/249/179	175/244/175	174/243/174
2190	f	175/244/175	179/249/179	180/250/180
	f	180/250/180	176/245/176	175/244/175
	f	116/251/116	120/256/120	181/257/181
	f	181/257/181	177/252/177	116/251/116
	f	177/252/177	181/257/181	182/258/182
2195	f	182/258/182	178/253/178	177/252/177
	f	178/253/178	182/258/182	183/259/183
	f	183/259/183	179/254/179	178/253/178
	f	179/254/179	183/259/183	184/260/184
	f	184/260/184	180/255/180	179/254/179
2200	f	120/256/120	124/261/124	185/262/185
	f	185/262/185	181/257/181	120/256/120
	f	181/257/181	185/262/185	186/263/186
	f	186/263/186	182/258/182	181/257/181
	f	182/258/182	186/263/186	187/264/187
2205	f	187/264/187	183/259/183	182/258/182
	f	183/259/183	187/264/187	188/265/188
	f	188/265/188	184/260/184	183/259/183
	f	124/261/124	128/266/128	189/267/189
	f	189/267/189	185/262/185	124/261/124
2210	f	185/262/185	189/267/189	190/268/190
	f	190/268/190	186/263/186	185/262/185
	f	186/263/186	190/268/190	191/269/191
	f	191/269/191	187/264/187	186/263/186
	f	187/264/187	191/269/191	192/270/192
2215	f	192/270/192	188/265/188	187/264/187
	f	128/266/128	132/271/132	193/272/193
	f	193/272/193	189/267/189	128/266/128
	f	189/267/189	193/272/193	194/273/194
	f	194/273/194	190/268/190	189/267/189
2220	f	190/268/190	194/273/194	195/274/195
	f	195/274/195	191/269/191	190/268/190
	f	191/269/191	195/274/195	196/275/196
	f	196/275/196	192/270/192	191/269/191
	f	132/276/132	136/281/136	197/282/197
2225	f	197/282/197	193/277/193	132/276/132
	f	193/277/193	197/282/197	198/283/198

	f	198/283/198	194/278/194	193/277/193
	f	194/278/194	198/283/198	199/284/199
	f	199/284/199	195/279/195	194/278/194
2230	f	195/279/195	199/284/199	200/285/200
	f	200/285/200	196/280/196	195/279/195
	f	136/281/136	140/286/140	201/287/201
	f	201/287/201	197/282/197	136/281/136
	f	197/282/197	201/287/201	202/288/202
2235	f	202/288/202	198/283/198	197/282/197
	f	198/283/198	202/288/202	203/289/203
	f	203/289/203	199/284/199	198/283/198
	f	199/284/199	203/289/203	204/290/204
	f	204/290/204	200/285/200	199/284/199
2240	f	140/286/140	144/291/144	205/292/205
	f	205/292/205	201/287/201	140/286/140
	f	201/287/201	205/292/205	206/293/206
	f	206/293/206	202/288/202	201/287/201
	f	202/288/202	206/293/206	207/294/207
2245	f	207/294/207	203/289/203	202/288/202
	f	203/289/203	207/294/207	208/295/208
	f	208/295/208	204/290/204	203/289/203
	f	144/291/144	84/296/84	145/297/145
	f	145/297/145	205/292/205	144/291/144
2250	f	205/292/205	145/297/145	146/298/146
	f	146/298/146	206/293/206	205/292/205
	f	206/293/206	146/298/146	147/299/147
	f	147/299/147	207/294/207	206/293/206
	f	207/294/207	147/299/147	148/300/148
2255	f	148/300/148	208/295/208	207/294/207
	f	148/301/148	152/306/152	213/307/213
	f	213/307/213	209/302/209	148/301/148
	f	209/302/209	213/307/213	214/308/214
	f	214/308/214	210/303/210	209/302/209
2260	f	210/303/210	214/308/214	215/309/215
	f	215/309/215	211/304/211	210/303/210
	f	211/304/211	215/309/215	212/310/212
	f	152/306/152	156/311/156	216/312/216
	f	216/312/216	213/307/213	152/306/152
2265	f	213/307/213	216/312/216	217/313/217
	f	217/313/217	214/308/214	213/307/213
	f	214/308/214	217/313/217	218/314/218
	f	218/314/218	215/309/215	214/308/214
	f	215/309/215	218/314/218	212/315/212
2270	f	156/311/156	160/316/160	219/317/219
	f	219/317/219	216/312/216	156/311/156
	f	216/312/216	219/317/219	220/318/220
	f	220/318/220	217/313/217	216/312/216
	f	217/313/217	220/318/220	221/319/221
2275	f	221/319/221	218/314/218	217/313/217
	f	218/314/218	221/319/221	212/320/212
	f	160/316/160	164/321/164	222/322/222
	f	222/322/222	219/317/219	160/316/160
	f	219/317/219	222/322/222	223/323/223
2280	f	223/323/223	220/318/220	219/317/219
	f	220/318/220	223/323/223	224/324/224
	f	224/324/224	221/319/221	220/318/220
	f	221/319/221	224/324/224	212/325/212

	f	164/326/164	168/331/168	225/332/225
2285	f	225/332/225	222/327/222	164/326/164
	f	222/327/222	225/332/225	226/333/226
	f	226/333/226	223/328/223	222/327/222
	f	223/328/223	226/333/226	227/334/227
	f	227/334/227	224/329/224	223/328/223
2290	f	224/329/224	227/334/227	212/335/212
	f	168/331/168	172/336/172	228/337/228
	f	228/337/228	225/332/225	168/331/168
	f	225/332/225	228/337/228	229/338/229
	f	229/338/229	226/333/226	225/332/225
2295	f	226/333/226	229/338/229	230/339/230
	f	230/339/230	227/334/227	226/333/226
	f	227/334/227	230/339/230	212/340/212
	f	172/336/172	176/341/176	231/342/231
	f	231/342/231	228/337/228	172/336/172
2300	f	228/337/228	231/342/231	232/343/232
	f	232/343/232	229/338/229	228/337/228
	f	229/338/229	232/343/232	233/344/233
	f	233/344/233	230/339/230	229/338/229
	f	230/339/230	233/344/233	212/345/212
2305	f	176/341/176	180/346/180	234/347/234
	f	234/347/234	231/342/231	176/341/176
	f	231/342/231	234/347/234	235/348/235
	f	235/348/235	232/343/232	231/342/231
	f	232/343/232	235/348/235	236/349/236
2310	f	236/349/236	233/344/233	232/343/232
	f	233/344/233	236/349/236	212/350/212
	f	180/351/180	184/356/184	237/357/237
	f	237/357/237	234/352/234	180/351/180
	f	234/352/234	237/357/237	238/358/238
2315	f	238/358/238	235/353/235	234/352/234
	f	235/353/235	238/358/238	239/359/239
	f	239/359/239	236/354/236	235/353/235
	f	236/354/236	239/359/239	212/360/212
	f	184/356/184	188/361/188	240/362/240
2320	f	240/362/240	237/357/237	184/356/184
	f	237/357/237	240/362/240	241/363/241
	f	241/363/241	238/358/238	237/357/237
	f	238/358/238	241/363/241	242/364/242
	f	242/364/242	239/359/239	238/358/238
2325	f	239/359/239	242/364/242	212/365/212
	f	188/361/188	192/366/192	243/367/243
	f	243/367/243	240/362/240	188/361/188
	f	240/362/240	243/367/243	244/368/244
	f	244/368/244	241/363/241	240/362/240
2330	f	241/363/241	244/368/244	245/369/245
	f	245/369/245	242/364/242	241/363/241
	f	242/364/242	245/369/245	212/370/212
	f	192/366/192	196/371/196	246/372/246
	f	246/372/246	243/367/243	192/366/192
2335	f	243/367/243	246/372/246	247/373/247
	f	247/373/247	244/368/244	243/367/243
	f	244/368/244	247/373/247	248/374/248
	f	248/374/248	245/369/245	244/368/244
	f	245/369/245	248/374/248	212/375/212
2340	f	196/376/196	200/381/200	249/382/249

	f	249/382/249	246/377/246	196/376/196
	f	246/377/246	249/382/249	250/383/250
	f	250/383/250	247/378/247	246/377/246
	f	247/378/247	250/383/250	251/384/251
2345	f	251/384/251	248/379/248	247/378/247
	f	248/379/248	251/384/251	212/385/212
	f	200/381/200	204/386/204	252/387/252
	f	252/387/252	249/382/249	200/381/200
	f	249/382/249	252/387/252	253/388/253
2350	f	253/388/253	250/383/250	249/382/249
	f	250/383/250	253/388/253	254/389/254
	f	254/389/254	251/384/251	250/383/250
	f	251/384/251	254/389/254	212/390/212
	f	204/386/204	208/391/208	255/392/255
2355	f	255/392/255	252/387/252	204/386/204
	f	252/387/252	255/392/255	256/393/256
	f	256/393/256	253/388/253	252/387/252
	f	253/388/253	256/393/256	257/394/257
	f	257/394/257	254/389/254	253/388/253
2360	f	254/389/254	257/394/257	212/395/212
	f	208/391/208	148/396/148	209/397/209
	f	209/397/209	255/392/255	208/391/208
	f	255/392/255	209/397/209	210/398/210
	f	210/398/210	256/393/256	255/392/255
2365	f	256/393/256	210/398/210	211/399/211
	f	211/399/211	257/394/257	256/393/256
	f	257/394/257	211/399/211	212/400/212
	f	258/401/258	263/406/263	264/407/264
	f	264/407/264	259/402/259	258/401/258
2370	f	259/402/259	264/407/264	265/408/265
	f	265/408/265	260/403/260	259/402/259
	f	260/403/260	265/408/265	266/409/266
	f	266/409/266	261/404/261	260/403/260
	f	261/404/261	266/409/266	267/410/267
2375	f	267/410/267	262/405/262	261/404/261
	f	263/406/263	268/411/268	269/412/269
	f	269/412/269	264/407/264	263/406/263
	f	264/407/264	269/412/269	270/413/270
	f	270/413/270	265/408/265	264/407/264
2380	f	265/408/265	270/413/270	271/414/271
	f	271/414/271	266/409/266	265/408/265
	f	266/409/266	271/414/271	272/415/272
	f	272/415/272	267/410/267	266/409/266
	f	268/411/268	273/416/273	274/417/274
2385	f	274/417/274	269/412/269	268/411/268
	f	269/412/269	274/417/274	275/418/275
	f	275/418/275	270/413/270	269/412/269
	f	270/413/270	275/418/275	276/419/276
	f	276/419/276	271/414/271	270/413/270
2390	f	271/414/271	276/419/276	277/420/277
	f	277/420/277	272/415/272	271/414/271
	f	273/416/273	278/421/278	279/422/279
	f	279/422/279	274/417/274	273/416/273
	f	274/417/274	279/422/279	280/423/280
2395	f	280/423/280	275/418/275	274/417/274
	f	275/418/275	280/423/280	281/424/281
	f	281/424/281	276/419/276	275/418/275

	f	276/419/276	281/424/281	282/425/282
	f	282/425/282	277/420/277	276/419/276
2400	f	278/426/278	283/431/283	284/432/284
	f	284/432/284	279/427/279	278/426/278
	f	279/427/279	284/432/284	285/433/285
	f	285/433/285	280/428/280	279/427/279
	f	280/428/280	285/433/285	286/434/286
2405	f	286/434/286	281/429/281	280/428/280
	f	281/429/281	286/434/286	287/435/287
	f	287/435/287	282/430/282	281/429/281
	f	283/431/283	288/436/288	289/437/289
	f	289/437/289	284/432/284	283/431/283
2410	f	284/432/284	289/437/289	290/438/290
	f	290/438/290	285/433/285	284/432/284
	f	285/433/285	290/438/290	291/439/291
	f	291/439/291	286/434/286	285/433/285
	f	286/434/286	291/439/291	292/440/292
2415	f	292/440/292	287/435/287	286/434/286
	f	288/436/288	293/441/293	294/442/294
	f	294/442/294	289/437/289	288/436/288
	f	289/437/289	294/442/294	295/443/295
	f	295/443/295	290/438/290	289/437/289
2420	f	290/438/290	295/443/295	296/444/296
	f	296/444/296	291/439/291	290/438/290
	f	291/439/291	296/444/296	297/445/297
	f	297/445/297	292/440/292	291/439/291
	f	293/441/293	258/446/258	259/447/259
2425	f	259/447/259	294/442/294	293/441/293
	f	294/442/294	259/447/259	260/448/260
	f	260/448/260	295/443/295	294/442/294
	f	295/443/295	260/448/260	261/449/261
	f	261/449/261	296/444/296	295/443/295
2430	f	296/444/296	261/449/261	262/450/262
	f	262/450/262	297/445/297	296/444/296
	f	262/451/262	267/456/267	302/457/302
	f	302/457/302	298/452/298	262/451/262
	f	298/452/298	302/457/302	303/458/303
2435	f	303/458/303	299/453/299	298/452/298
	f	299/453/299	303/458/303	304/459/304
	f	304/459/304	300/454/300	299/453/299
	f	300/454/300	304/459/304	305/460/305
	f	305/460/305	301/455/301	300/454/300
2440	f	267/456/267	272/461/272	306/462/306
	f	306/462/306	302/457/302	267/456/267
	f	302/457/302	306/462/306	307/463/307
	f	307/463/307	303/458/303	302/457/302
	f	303/458/303	307/463/307	308/464/308
2445	f	308/464/308	304/459/304	303/458/303
	f	304/459/304	308/464/308	309/465/309
	f	309/465/309	305/460/305	304/459/304
	f	272/461/272	277/466/277	310/467/310
	f	310/467/310	306/462/306	272/461/272
2450	f	306/462/306	310/467/310	311/468/311
	f	311/468/311	307/463/307	306/462/306
	f	307/463/307	311/468/311	312/469/312
	f	312/469/312	308/464/308	307/463/307
	f	308/464/308	312/469/312	313/470/313

2455 f 313/470/313 309/465/309 308/464/308
 f 277/466/277 282/471/282 314/472/314
 f 314/472/314 310/467/310 277/466/277
 f 310/467/310 314/472/314 315/473/315
 f 315/473/315 311/468/311 310/467/310
 2460 f 311/468/311 315/473/315 316/474/316
 f 316/474/316 312/469/312 311/468/311
 f 312/469/312 316/474/316 317/475/317
 f 317/475/317 313/470/313 312/469/312
 f 282/476/282 287/481/287 318/482/318
 2465 f 318/482/318 314/477/314 282/476/282
 f 314/477/314 318/482/318 319/483/319
 f 319/483/319 315/478/315 314/477/314
 f 315/478/315 319/483/319 320/484/320
 f 320/484/320 316/479/316 315/478/315
 2470 f 316/479/316 320/484/320 321/485/321
 f 321/485/321 317/480/317 316/479/316
 f 287/481/287 292/486/292 322/487/322
 f 322/487/322 318/482/318 287/481/287
 f 318/482/318 322/487/322 323/488/323
 2475 f 323/488/323 319/483/319 318/482/318
 f 319/483/319 323/488/323 324/489/324
 f 324/489/324 320/484/320 319/483/319
 f 320/484/320 324/489/324 325/490/325
 f 325/490/325 321/485/321 320/484/320
 2480 f 292/486/292 297/491/297 326/492/326
 f 326/492/326 322/487/322 292/486/292
 f 322/487/322 326/492/326 327/493/327
 f 327/493/327 323/488/323 322/487/322
 f 323/488/323 327/493/327 328/494/328
 2485 f 328/494/328 324/489/324 323/488/323
 f 324/489/324 328/494/328 329/495/329
 f 329/495/329 325/490/325 324/489/324
 f 297/491/297 262/496/262 298/497/298
 f 298/497/298 326/492/326 297/491/297
 2490 f 326/492/326 298/497/298 299/498/299
 f 299/498/299 327/493/327 326/492/326
 f 327/493/327 299/498/299 300/499/300
 f 300/499/300 328/494/328 327/493/327
 f 328/494/328 300/499/300 301/500/301
 2495 f 301/500/301 329/495/329 328/494/328
 f 330/501/330 335/506/335 336/507/336
 f 336/507/336 331/502/331 330/501/330
 f 331/502/331 336/507/336 337/508/337
 f 337/508/337 332/503/332 331/502/331
 2500 f 332/503/332 337/508/337 338/509/338
 f 338/509/338 333/504/333 332/503/332
 f 333/504/333 338/509/338 339/510/339
 f 339/510/339 334/505/334 333/504/333
 f 335/506/335 340/511/340 341/512/341
 2505 f 341/512/341 336/507/336 335/506/335
 f 336/507/336 341/512/341 342/513/342
 f 342/513/342 337/508/337 336/507/336
 f 337/508/337 342/513/342 343/514/343
 f 343/514/343 338/509/338 337/508/337
 2510 f 338/509/338 343/514/343 344/515/344
 f 344/515/344 339/510/339 338/509/338

	f	340/511/340	345/516/345	346/517/346
	f	346/517/346	341/512/341	340/511/340
	f	341/512/341	346/517/346	347/518/347
2515	f	347/518/347	342/513/342	341/512/341
	f	342/513/342	347/518/347	348/519/348
	f	348/519/348	343/514/343	342/513/342
	f	343/514/343	348/519/348	349/520/349
	f	349/520/349	344/515/344	343/514/343
2520	f	345/516/345	350/521/350	351/522/351
	f	351/522/351	346/517/346	345/516/345
	f	346/517/346	351/522/351	352/523/352
	f	352/523/352	347/518/347	346/517/346
	f	347/518/347	352/523/352	353/524/353
2525	f	353/524/353	348/519/348	347/518/347
	f	348/519/348	353/524/353	354/525/354
	f	354/525/354	349/520/349	348/519/348
	f	350/526/350	355/531/355	356/532/356
	f	356/532/356	351/527/351	350/526/350
2530	f	351/527/351	356/532/356	357/533/357
	f	357/533/357	352/528/352	351/527/351
	f	352/528/352	357/533/357	358/534/358
	f	358/534/358	353/529/353	352/528/352
	f	353/529/353	358/534/358	359/535/359
2535	f	359/535/359	354/530/354	353/529/353
	f	355/531/355	360/536/360	361/537/361
	f	361/537/361	356/532/356	355/531/355
	f	356/532/356	361/537/361	362/538/362
	f	362/538/362	357/533/357	356/532/356
2540	f	357/533/357	362/538/362	363/539/363
	f	363/539/363	358/534/358	357/533/357
	f	358/534/358	363/539/363	364/540/364
	f	364/540/364	359/535/359	358/534/358
	f	360/536/360	365/541/365	366/542/366
2545	f	366/542/366	361/537/361	360/536/360
	f	361/537/361	366/542/366	367/543/367
	f	367/543/367	362/538/362	361/537/361
	f	362/538/362	367/543/367	368/544/368
	f	368/544/368	363/539/363	362/538/362
2550	f	363/539/363	368/544/368	369/545/369
	f	369/545/369	364/540/364	363/539/363
	f	365/541/365	330/546/330	331/547/331
	f	331/547/331	366/542/366	365/541/365
	f	366/542/366	331/547/331	332/548/332
2555	f	332/548/332	367/543/367	366/542/366
	f	367/543/367	332/548/332	333/549/333
	f	333/549/333	368/544/368	367/543/367
	f	368/544/368	333/549/333	334/550/334
	f	334/550/334	369/545/369	368/544/368
2560	f	334/551/334	339/556/339	374/557/374
	f	374/557/374	370/552/370	334/551/334
	f	370/552/370	374/557/374	375/558/375
	f	375/558/375	371/553/371	370/552/370
	f	371/553/371	375/558/375	376/559/376
2565	f	376/559/376	372/554/372	371/553/371
	f	372/554/372	376/559/376	377/560/377
	f	377/560/377	373/555/373	372/554/372
	f	339/556/339	344/561/344	378/562/378

	f	378/562/378	374/557/374	339/556/339
2570	f	374/557/374	378/562/378	379/563/379
	f	379/563/379	375/558/375	374/557/374
	f	375/558/375	379/563/379	380/564/380
	f	380/564/380	376/559/376	375/558/375
	f	376/559/376	380/564/380	381/565/381
2575	f	381/565/381	377/560/377	376/559/376
	f	344/561/344	349/566/349	382/567/382
	f	382/567/382	378/562/378	344/561/344
	f	378/562/378	382/567/382	383/568/383
	f	383/568/383	379/563/379	378/562/378
2580	f	379/563/379	383/568/383	384/569/384
	f	384/569/384	380/564/380	379/563/379
	f	380/564/380	384/569/384	385/570/385
	f	385/570/385	381/565/381	380/564/380
	f	349/566/349	354/571/354	386/572/386
2585	f	386/572/386	382/567/382	349/566/349
	f	382/567/382	386/572/386	387/573/387
	f	387/573/387	383/568/383	382/567/382
	f	383/568/383	387/573/387	388/574/388
	f	388/574/388	384/569/384	383/568/383
2590	f	384/569/384	388/574/388	389/575/389
	f	389/575/389	385/570/385	384/569/384
	f	354/576/354	359/581/359	390/582/390
	f	390/582/390	386/577/386	354/576/354
	f	386/577/386	390/582/390	391/583/391
2595	f	391/583/391	387/578/387	386/577/386
	f	387/578/387	391/583/391	392/584/392
	f	392/584/392	388/579/388	387/578/387
	f	388/579/388	392/584/392	393/585/393
	f	393/585/393	389/580/389	388/579/388
2600	f	359/581/359	364/586/364	394/587/394
	f	394/587/394	390/582/390	359/581/359
	f	390/582/390	394/587/394	395/588/395
	f	395/588/395	391/583/391	390/582/390
	f	391/583/391	395/588/395	396/589/396
2605	f	396/589/396	392/584/392	391/583/391
	f	392/584/392	396/589/396	397/590/397
	f	397/590/397	393/585/393	392/584/392
	f	364/586/364	369/591/369	398/592/398
	f	398/592/398	394/587/394	364/586/364
2610	f	394/587/394	398/592/398	399/593/399
	f	399/593/399	395/588/395	394/587/394
	f	395/588/395	399/593/399	400/594/400
	f	400/594/400	396/589/396	395/588/395
	f	396/589/396	400/594/400	401/595/401
2615	f	401/595/401	397/590/397	396/589/396
	f	369/591/369	334/596/334	370/597/370
	f	370/597/370	398/592/398	369/591/369
	f	398/592/398	370/597/370	371/598/371
	f	371/598/371	399/593/399	398/592/398
2620	f	399/593/399	371/598/371	372/599/372
	f	372/599/372	400/594/400	399/593/399
	f	400/594/400	372/599/372	373/600/373
	f	373/600/373	401/595/401	400/594/400
	f	407/607/407	403/602/403	402/601/402
2625	f	403/602/403	407/607/407	408/608/408

	f	408/608/408	404/603/404	403/602/403
	f	404/603/404	408/608/408	409/609/409
	f	409/609/409	405/604/405	404/603/404
	f	405/604/405	409/609/409	410/610/410
2630	f	410/610/410	406/605/406	405/604/405
	f	411/612/411	407/607/407	402/606/402
	f	407/607/407	411/612/411	412/613/412
	f	412/613/412	408/608/408	407/607/407
	f	408/608/408	412/613/412	413/614/413
2635	f	413/614/413	409/609/409	408/608/408
	f	409/609/409	413/614/413	414/615/414
	f	414/615/414	410/610/410	409/609/409
	f	415/617/415	411/612/411	402/611/402
	f	411/612/411	415/617/415	416/618/416
2640	f	416/618/416	412/613/412	411/612/411
	f	412/613/412	416/618/416	417/619/417
	f	417/619/417	413/614/413	412/613/412
	f	413/614/413	417/619/417	418/620/418
	f	418/620/418	414/615/414	413/614/413
2645	f	419/622/419	415/617/415	402/616/402
	f	415/617/415	419/622/419	420/623/420
	f	420/623/420	416/618/416	415/617/415
	f	416/618/416	420/623/420	421/624/421
	f	421/624/421	417/619/417	416/618/416
2650	f	417/619/417	421/624/421	422/625/422
	f	422/625/422	418/620/418	417/619/417
	f	423/632/423	419/627/419	402/626/402
	f	419/627/419	423/632/423	424/633/424
	f	424/633/424	420/628/420	419/627/419
2655	f	420/628/420	424/633/424	425/634/425
	f	425/634/425	421/629/421	420/628/420
	f	421/629/421	425/634/425	426/635/426
	f	426/635/426	422/630/422	421/629/421
	f	427/637/427	423/632/423	402/631/402
2660	f	423/632/423	427/637/427	428/638/428
	f	428/638/428	424/633/424	423/632/423
	f	424/633/424	428/638/428	429/639/429
	f	429/639/429	425/634/425	424/633/424
	f	425/634/425	429/639/429	430/640/430
2665	f	430/640/430	426/635/426	425/634/425
	f	431/642/431	427/637/427	402/636/402
	f	427/637/427	431/642/431	432/643/432
	f	432/643/432	428/638/428	427/637/427
	f	428/638/428	432/643/432	433/644/433
2670	f	433/644/433	429/639/429	428/638/428
	f	429/639/429	433/644/433	434/645/434
	f	434/645/434	430/640/430	429/639/429
	f	435/647/435	431/642/431	402/641/402
	f	431/642/431	435/647/435	436/648/436
2675	f	436/648/436	432/643/432	431/642/431
	f	432/643/432	436/648/436	437/649/437
	f	437/649/437	433/644/433	432/643/432
	f	433/644/433	437/649/437	438/650/438
	f	438/650/438	434/645/434	433/644/433
2680	f	439/652/439	435/652/435	402/651/402
	f	435/652/435	439/657/439	440/658/440
	f	440/658/440	436/653/436	435/652/435

	f	436/653/436	440/658/440	441/659/441
	f	441/659/441	437/654/437	436/653/436
2685	f	437/654/437	441/659/441	442/660/442
	f	442/660/442	438/655/438	437/654/437
	f	443/662/443	439/657/439	402/656/402
	f	439/657/439	443/662/443	444/663/444
	f	444/663/444	440/658/440	439/657/439
2690	f	440/658/440	444/663/444	445/664/445
	f	445/664/445	441/659/441	440/658/440
	f	441/659/441	445/664/445	446/665/446
	f	446/665/446	442/660/442	441/659/441
	f	447/667/447	443/662/443	402/661/402
2695	f	443/662/443	447/667/447	448/668/448
	f	448/668/448	444/663/444	443/662/443
	f	444/663/444	448/668/448	449/669/449
	f	449/669/449	445/664/445	444/663/444
	f	445/664/445	449/669/449	450/670/450
2700	f	450/670/450	446/665/446	445/664/445
	f	451/672/451	447/667/447	402/666/402
	f	447/667/447	451/672/451	452/673/452
	f	452/673/452	448/668/448	447/667/447
	f	448/668/448	452/673/452	453/674/453
2705	f	453/674/453	449/669/449	448/668/448
	f	449/669/449	453/674/453	454/675/454
	f	454/675/454	450/670/450	449/669/449
	f	455/682/455	451/677/451	402/676/402
	f	451/677/451	455/682/455	456/683/456
2710	f	456/683/456	452/678/452	451/677/451
	f	452/678/452	456/683/456	457/684/457
	f	457/684/457	453/679/453	452/678/452
	f	453/679/453	457/684/457	458/685/458
	f	458/685/458	454/680/454	453/679/453
2715	f	459/687/459	455/682/455	402/681/402
	f	455/682/455	459/687/459	460/688/460
	f	460/688/460	456/683/456	455/682/455
	f	456/683/456	460/688/460	461/689/461
	f	461/689/461	457/684/457	456/683/456
2720	f	457/684/457	461/689/461	462/690/462
	f	462/690/462	458/685/458	457/684/457
	f	463/692/463	459/687/459	402/686/402
	f	459/687/459	463/692/463	464/693/464
	f	464/693/464	460/688/460	459/687/459
2725	f	460/688/460	464/693/464	465/694/465
	f	465/694/465	461/689/461	460/688/460
	f	461/689/461	465/694/465	466/695/466
	f	466/695/466	462/690/462	461/689/461
	f	403/697/403	463/692/463	402/691/402
2730	f	463/692/463	403/697/403	404/698/404
	f	404/698/404	464/693/464	463/692/463
	f	464/693/464	404/698/404	405/699/405
	f	405/699/405	465/694/465	464/693/464
	f	465/694/465	405/699/405	406/700/406
2735	f	406/700/406	466/695/466	465/694/465
	f	406/701/406	410/706/410	471/707/471
	f	471/707/471	467/702/467	406/701/406
	f	467/702/467	471/707/471	472/708/472
	f	472/708/472	468/703/468	467/702/467

```

2740  f 468/703/468 472/708/472 473/709/473
      f 473/709/473 469/704/469 468/703/468
      f 469/704/469 473/709/473 474/710/474
      f 474/710/474 470/705/470 469/704/469
      f 410/706/410 414/711/414 475/712/475
2745  f 475/712/475 471/707/471 410/706/410
      f 471/707/471 475/712/475 476/713/476
      f 476/713/476 472/708/472 471/707/471
      f 472/708/472 476/713/476 477/714/477
      f 477/714/477 473/709/473 472/708/472
2750  f 473/709/473 477/714/477 478/715/478
      f 478/715/478 474/710/474 473/709/473
      f 414/711/414 418/716/418 479/717/479
      f 479/717/479 475/712/475 414/711/414
      f 475/712/475 479/717/479 480/718/480
2755  f 480/718/480 476/713/476 475/712/475
      f 476/713/476 480/718/480 481/719/481
      f 481/719/481 477/714/477 476/713/476
      f 477/714/477 481/719/481 482/720/482
      f 482/720/482 478/715/478 477/714/477
2760  f 418/716/418 422/721/422 483/722/483
      f 483/722/483 479/717/479 418/716/418
      f 479/717/479 483/722/483 484/723/484
      f 484/723/484 480/718/480 479/717/479
      f 480/718/480 484/723/484 485/724/485
2765  f 485/724/485 481/719/481 480/718/480
      f 481/719/481 485/724/485 486/725/486
      f 486/725/486 482/720/482 481/719/481
      f 422/726/422 426/731/426 487/732/487
      f 487/732/487 483/727/483 422/726/422
2770  f 483/727/483 487/732/487 488/733/488
      f 488/733/488 484/728/484 483/727/483
      f 484/728/484 488/733/488 489/734/489
      f 489/734/489 485/729/485 484/728/484
      f 485/729/485 489/734/489 490/735/490
2775  f 490/735/490 486/730/486 485/729/485
      f 426/731/426 430/736/430 491/737/491
      f 491/737/491 487/732/487 426/731/426
      f 487/732/487 491/737/491 492/738/492
      f 492/738/492 488/733/488 487/732/487
2780  f 488/733/488 492/738/492 493/739/493
      f 493/739/493 489/734/489 488/733/488
      f 489/734/489 493/739/493 494/740/494
      f 494/740/494 490/735/490 489/734/489
      f 430/736/430 434/741/434 495/742/495
2785  f 495/742/495 491/737/491 430/736/430
      f 491/737/491 495/742/495 496/743/496
      f 496/743/496 492/738/492 491/737/491
      f 492/738/492 496/743/496 497/744/497
      f 497/744/497 493/739/493 492/738/492
2790  f 493/739/493 497/744/497 498/745/498
      f 498/745/498 494/740/494 493/739/493
      f 434/741/434 438/746/438 499/747/499
      f 499/747/499 495/742/495 434/741/434
      f 495/742/495 499/747/499 500/748/500
2795  f 500/748/500 496/743/496 495/742/495
      f 496/743/496 500/748/500 501/749/501

```


	f	501/749/501	497/744/497	496/743/496
	f	497/744/497	501/749/501	502/750/502
	f	502/750/502	498/745/498	497/744/497
2800	f	438/751/438	442/756/442	503/757/503
	f	503/757/503	499/752/499	438/751/438
	f	499/752/499	503/757/503	504/758/504
	f	504/758/504	500/753/500	499/752/499
	f	500/753/500	504/758/504	505/759/505
2805	f	505/759/505	501/754/501	500/753/500
	f	501/754/501	505/759/505	506/760/506
	f	506/760/506	502/755/502	501/754/501
	f	442/756/442	446/761/446	507/762/507
	f	507/762/507	503/757/503	442/756/442
2810	f	503/757/503	507/762/507	508/763/508
	f	508/763/508	504/758/504	503/757/503
	f	504/758/504	508/763/508	509/764/509
	f	509/764/509	505/759/505	504/758/504
	f	505/759/505	509/764/509	510/765/510
2815	f	510/765/510	506/760/506	505/759/505
	f	446/761/446	450/766/450	511/767/511
	f	511/767/511	507/762/507	446/761/446
	f	507/762/507	511/767/511	512/768/512
	f	512/768/512	508/763/508	507/762/507
2820	f	508/763/508	512/768/512	513/769/513
	f	513/769/513	509/764/509	508/763/508
	f	509/764/509	513/769/513	514/770/514
	f	514/770/514	510/765/510	509/764/509
	f	450/766/450	454/771/454	515/772/515
2825	f	515/772/515	511/767/511	450/766/450
	f	511/767/511	515/772/515	516/773/516
	f	516/773/516	512/768/512	511/767/511
	f	512/768/512	516/773/516	517/774/517
	f	517/774/517	513/769/513	512/768/512
2830	f	513/769/513	517/774/517	518/775/518
	f	518/775/518	514/770/514	513/769/513
	f	454/776/454	458/781/458	519/782/519
	f	519/782/519	515/777/515	454/776/454
	f	515/777/515	519/782/519	520/783/520
2835	f	520/783/520	516/778/516	515/777/515
	f	516/778/516	520/783/520	521/784/521
	f	521/784/521	517/779/517	516/778/516
	f	517/779/517	521/784/521	522/785/522
	f	522/785/522	518/780/518	517/779/517
2840	f	458/781/458	462/786/462	523/787/523
	f	523/787/523	519/782/519	458/781/458
	f	519/782/519	523/787/523	524/788/524
	f	524/788/524	520/783/520	519/782/519
	f	520/783/520	524/788/524	525/789/525
2845	f	525/789/525	521/784/521	520/783/520
	f	521/784/521	525/789/525	526/790/526
	f	526/790/526	522/785/522	521/784/521
	f	462/786/462	466/791/466	527/792/527
	f	527/792/527	523/787/523	462/786/462
2850	f	523/787/523	527/792/527	528/793/528
	f	528/793/528	524/788/524	523/787/523
	f	524/788/524	528/793/528	529/794/529
	f	529/794/529	525/789/525	524/788/524

```

2855 f 525/789/525 529/794/529 530/795/530
f 530/795/530 526/790/526 525/789/525
f 466/791/466 406/796/406 467/797/467
f 467/797/467 527/792/527 466/791/466
f 527/792/527 467/797/467 468/798/468
f 468/798/468 528/793/528 527/792/527
2860 f 528/793/528 468/798/468 469/799/469
f 469/799/469 529/794/529 528/793/528
f 529/794/529 469/799/469 470/800/470
f 470/800/470 530/795/530 529/794/529
# 992 faces
2865
g

```

39 prototype/vec3-add.pd

```

#N canvas 0 0 450 300 10;
#X obj 24 21 inlet;
#X obj 24 42 unpack f f f;
#X obj 24 84 pack f f f;
5 #X obj 24 105 outlet;
#X obj 107 39 unpack f f f;
#X obj 109 17 inlet;
#X obj 24 63 + \$1;
#X obj 54 63 + \$2;
10 #X obj 83 63 + \$3;
#X connect 0 0 1 0;
#X connect 1 0 6 0;
#X connect 1 1 7 0;
#X connect 1 2 8 0;
15 #X connect 2 0 3 0;
#X connect 4 0 6 1;
#X connect 4 1 7 1;
#X connect 4 2 8 1;
#X connect 5 0 4 0;
20 #X connect 6 0 2 0;
#X connect 7 0 2 1;
#X connect 8 0 2 2;

```

40 prototype/vec3-length.pd

```

#N canvas 0 0 450 300 10;
#X obj 23 22 inlet;
#X obj 23 43 unpack f f f;
#X obj 23 64 t f f;
5 #X obj 23 85 *;
#X obj 61 64 t f f;
#X obj 61 85 *;
#X obj 103 65 t f f;
#X obj 103 86 *;
10 #X obj 24 106 +;
#X obj 61 106 +;
#X obj 24 127 sqrt;
#X obj 24 148 outlet;
#X connect 0 0 1 0;
15 #X connect 1 0 2 0;

```

```

#X connect 1 1 4 0;
#X connect 1 2 6 0;
#X connect 2 0 3 0;
#X connect 2 1 3 1;
20 #X connect 3 0 8 0;
#X connect 4 0 5 0;
#X connect 4 1 5 1;
#X connect 5 0 9 0;
#X connect 6 0 7 0;
25 #X connect 6 1 7 1;
#X connect 7 0 9 1;
#X connect 8 0 10 0;
#X connect 9 0 8 1;
#X connect 10 0 11 0;

```

41 prototype/vec3-normalize.pd

```

#N canvas 0 0 450 300 10;
#X obj 21 22 inlet;
#X obj 21 43 t a a;
#X obj 23 168 outlet;
5 #X obj 48 65 vec3-length;
#X obj 48 86 swap 1;
#X obj 48 107 /;
#X obj 24 137 vec3-scale;
#X connect 0 0 1 0;
10 #X connect 1 0 6 0;
#X connect 1 1 3 0;
#X connect 3 0 4 0;
#X connect 4 0 5 0;
#X connect 4 1 5 1;
15 #X connect 5 0 6 1;
#X connect 6 0 2 0;

```

42 prototype/vec3-scale.pd

```

#N canvas 0 0 450 300 10;
#X obj 24 21 inlet;
#X obj 24 42 unpack f f f;
#X obj 24 84 pack f f f;
5 #X obj 24 105 outlet;
#X obj 77 22 inlet;
#X obj 24 63 * \$1;
#X obj 54 63 * \$1;
#X obj 83 63 * \$1;
10 #X connect 0 0 1 0;
#X connect 1 0 5 0;
#X connect 1 1 6 0;
#X connect 1 2 7 0;
#X connect 2 0 3 0;
15 #X connect 4 0 5 1;
#X connect 4 0 6 1;
#X connect 4 0 7 1;
#X connect 5 0 2 0;
#X connect 6 0 2 1;
20 #X connect 7 0 2 2;

```

43 prototype/with-blend.pd

```

#N canvas 0 0 450 300 10;
#X obj 22 22 inlet;
#X obj 22 43 t a a a;
#X obj 41 90 outlet;
5 #X obj 62 68 GEMglEnable GL_BLEND;
#X obj 22 113 GEMglDisable GL_BLEND;
#X connect 0 0 1 0;
#X connect 1 0 4 0;
#X connect 1 1 2 0;
10 #X connect 1 2 3 0;

```

44 prototype/with-lighting.pd

```

#N canvas 0 0 450 300 10;
#X obj 22 22 inlet;
#X obj 22 43 t a a a;
#X obj 61 68 GEMglEnable GL_LIGHTING;
5 #X obj 22 113 GEMglDisable GL_LIGHTING;
#X obj 41 90 outlet;
#X connect 0 0 1 0;
#X connect 1 0 3 0;
#X connect 1 1 4 0;
10 #X connect 1 2 2 0;

```

45 prototype/without-lighting.pd

```

#N canvas 0 0 450 300 10;
#X obj 22 22 inlet;
#X obj 22 43 t a a a;
#X obj 41 90 outlet;
5 #X obj 61 68 GEMglDisable GL_LIGHTING;
#X obj 22 113 GEMglEnable GL_LIGHTING;
#X connect 0 0 1 0;
#X connect 1 0 4 0;
#X connect 1 1 2 0;
10 #X connect 1 2 3 0;

```

46 prototype/worm-sound.pd

```

#N canvas 363 530 768 343 10;
#X obj 23 9 inlet;
#X obj 20 71 t a a b;
#X obj 139 92 vec3-add;
5 #X obj 131 20 inlet;
#X obj 139 113 vec3-length;
#X obj 493 110 random 36;
#X obj 493 131 + 48;
#X obj 555 110 random 36;
10 #X msg 373 192 \$1 \, \$2 \$3;
#X obj 373 213 vline~;
#X obj 373 234 mtof~;
#X obj 373 255 osc~;
#X obj 172 253 vline~;

```

```
15  #X obj 622 133 + 10;
    #X obj 26 280 outlet;
    #X obj 23 30 route \$2;
    #X obj 127 278 outlet;
    #X obj 241 310 throw~ \$1-l;
20  #X obj 319 310 throw~ \$1-r;
    #X obj 148 165 expr 900 / ($f1 * $f1 + 0.001);
    #X msg 100 70 -1 -6 \$1;
    #X msg 226 63 1 -6 \$1;
    #X obj 151 145 t f f;
25  #X obj 319 289 *~;
    #X obj 241 289 *~;
    #X obj 16 95 vec3-add;
    #X obj 16 116 vec3-length;
    #X obj 49 256 vline~;
30  #X obj 25 167 expr 900 / ($f1 * $f1 + 0.001);
    #X obj 28 148 t f f;
    #X obj 25 188 clip 0 1;
    #X obj 148 189 clip 0 1;
    #X obj 139 134 + 30;
35  #X obj 16 137 + 30;
    #X obj 22 50 vec3-scale 4;
    #X obj 146 42 route float;
    #X obj 492 89 t b b b;
    #X obj 493 152 pack f f f;
40  #X obj 331 107 t f b;
    #X obj 371 108 random 48;
    #X obj 402 24 symbol;
    #X obj 327 86 f 60;
    #X obj 371 139 + 60;
45  #X msg 49 235 \$1 \, 0 200;
    #X msg 172 232 \$1 \, 0 200;
    #X obj 332 169 pack f f 200;
    #X obj 622 112 random 300;
    #X obj 555 131 + 36;
50  #X obj 328 44 symbol wriggle;
    #X obj 327 65 sel column wriggle;
    #X connect 0 0 15 0;
    #X connect 1 0 25 0;
    #X connect 1 1 2 0;
55  #X connect 1 2 48 0;
    #X connect 2 0 4 0;
    #X connect 3 0 35 0;
    #X connect 3 0 16 0;
    #X connect 4 0 32 0;
60  #X connect 5 0 6 0;
    #X connect 6 0 37 0;
    #X connect 7 0 47 0;
    #X connect 8 0 9 0;
    #X connect 9 0 10 0;
65  #X connect 10 0 11 0;
    #X connect 11 0 23 0;
    #X connect 11 0 24 0;
    #X connect 12 0 23 1;
    #X connect 13 0 37 2;
70  #X connect 15 0 34 0;
    #X connect 15 1 14 0;
```

```

#X connect 19 0 31 0;
#X connect 20 0 25 1;
#X connect 21 0 2 1;
75 #X connect 22 0 19 0;
#X connect 23 0 18 0;
#X connect 24 0 17 0;
#X connect 25 0 26 0;
#X connect 26 0 33 0;
80 #X connect 27 0 24 1;
#X connect 28 0 30 0;
#X connect 29 0 28 0;
#X connect 30 0 43 0;
#X connect 31 0 44 0;
85 #X connect 32 0 22 0;
#X connect 33 0 29 0;
#X connect 34 0 1 0;
#X connect 35 0 20 0;
#X connect 35 0 21 0;
90 #X connect 35 1 40 0;
#X connect 36 0 5 0;
#X connect 36 1 7 0;
#X connect 36 2 46 0;
#X connect 37 0 8 0;
95 #X connect 38 0 45 0;
#X connect 38 1 39 0;
#X connect 39 0 42 0;
#X connect 40 0 48 1;
#X connect 41 0 38 0;
100 #X connect 42 0 45 1;
#X connect 43 0 27 0;
#X connect 44 0 12 0;
#X connect 45 0 8 0;
#X connect 46 0 13 0;
105 #X connect 47 0 37 1;
#X connect 48 0 49 0;
#X connect 49 0 41 0;
#X connect 49 1 36 0;

```

47 prototype/worm-sounds.pd

```

#N canvas 0 0 450 300 10;
#X obj 159 130 catch~ \ $0-l;
#X obj 259 128 catch~ \ $0-r;
#X obj 160 233 outlet~;
5 #X obj 260 236 outlet~;
#X obj 127 17 inlet;
#X obj 37 18 inlet;
#X obj 159 151 expr~ 4*tanh($v1);
#X obj 259 154 expr~ 4*tanh($v1);
10 #X obj 39 270 worm-sound \ $0 12;
#X obj 39 210 worm-sound \ $0 9;
#X obj 39 50 worm-sound \ $0 1;
#X obj 39 70 worm-sound \ $0 2;
#X obj 39 90 worm-sound \ $0 3;
15 #X obj 39 110 worm-sound \ $0 4;
#X obj 39 130 worm-sound \ $0 5;
#X obj 39 150 worm-sound \ $0 6;

```

```

#X obj 39 170 worm-sound \ $0 7;
#X obj 39 190 worm-sound \ $0 8;
20 #X obj 39 230 worm-sound \ $0 10;
#X obj 39 250 worm-sound \ $0 11;
#X connect 0 0 6 0;
#X connect 1 0 7 0;
#X connect 4 0 10 1;
25 #X connect 5 0 10 0;
#X connect 6 0 2 0;
#X connect 7 0 3 0;
#X connect 9 0 18 0;
#X connect 9 1 18 1;
30 #X connect 10 0 11 0;
#X connect 10 1 11 1;
#X connect 11 0 12 0;
#X connect 11 1 12 1;
#X connect 12 0 13 0;
35 #X connect 12 1 13 1;
#X connect 13 0 14 0;
#X connect 13 1 14 1;
#X connect 14 0 15 0;
#X connect 14 1 15 1;
40 #X connect 15 0 16 0;
#X connect 15 1 16 1;
#X connect 16 0 17 0;
#X connect 16 1 17 1;
#X connect 17 0 9 0;
45 #X connect 17 1 9 1;
#X connect 18 0 19 0;
#X connect 18 1 19 1;
#X connect 19 0 8 0;
#X connect 19 1 8 1;

```

48 sketches/byzantine/byzantine.cc

```

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
5 #include <GL/glew.h>
#include <GLFW/glfw3.h>

#include <iostream>
#include <glm/glm.hpp>
10 #include <glm/gtc/matrix_inverse.hpp>
#include <glm/gtc/matrix_transform.hpp>

const int width = 640, height = 360;

15 const double twopi = 6.283185307179586;

const GLchar *vert_src =
    "#version 330 core\n"
    "uniform mat4 mvp;\n"
20 "in vec3 vertex;\n"
    "out vec3 colour;\n"
    "void main() { vec4 v = mvp * vec4(vertex, 1.0); v.w += 0.5 - v.z; gl_Position =

```

```

        ↵ = v; colour = vertex; }\n"
;

25  const GLchar *frag_src =
    "#version 330 core\n"
    "in vec3 colour;\n"
    "void main() { float edge = length(colour) > 1.25 ? 1.0 : 0.5; gl_FragData[0] ↵
        ↵ = vec4((0.5 * colour + vec3(0.5)) * edge, 1.0); }\n"
;

30  typedef struct {
    GLuint vbo;
    GLuint vao;
    GLint.mvp;
35  } G;

void init(G *g) {
    glGenBuffers(1, &g->vbo);
    glGenVertexArrays(1, &g->vao);
40  glBindVertexArray(g->vao);
    glBindBuffer(GL_ARRAY_BUFFER, g->vbo);
    GLfloat cube[] =
        { -1, -1, -1, -1, -1, 1, -1, 1, 1
          , -1, -1, -1, -1, -1, 1, 1, -1, 1
45  , -1, -1, -1, -1, 1, -1, -1, 1, 1
          , -1, -1, -1, -1, 1, -1, 1, 1, -1
          , -1, -1, -1, 1, -1, -1, 1, -1, 1
          , -1, -1, -1, 1, -1, -1, 1, 1, -1
          , 1, 1, 1, -1, -1, 1, -1, 1, 1
50  , 1, 1, 1, -1, -1, 1, 1, -1, 1
          , 1, 1, 1, -1, 1, -1, -1, 1, 1
          , 1, 1, 1, -1, 1, -1, 1, 1, -1
          , 1, 1, 1, 1, -1, -1, 1, -1, 1
          , 1, 1, 1, 1, -1, -1, 1, 1, -1
55  };
    glBufferData(GL_ARRAY_BUFFER, sizeof(cube), cube, GL_STATIC_DRAW);
    GLuint prog = glCreateProgram();
    GLuint vert = glCreateShader(GL_VERTEX_SHADER);
    glShaderSource(vert, 1, &vert_src, 0);
60  glCompileShader(vert);
    glAttachShader(prog, vert);
    GLuint frag = glCreateShader(GL_FRAGMENT_SHADER);
    glShaderSource(frag, 1, &frag_src, 0);
    glCompileShader(frag);
65  glAttachShader(prog, frag);
    glLinkProgram(prog);
    glUseProgram(prog);
    g->mvp = glGetUniformLocation(prog, "mvp");
    glVertexAttribPointer(glGetAttribLocation(prog, "vertex"), 3, GL_FLOAT, ↵
        ↵ GL_FALSE, 0, 0);
70  glEnableVertexAttribArray(0);
    glEnable(GL_DEPTH_TEST);
}

glm::mat4 byzantine(float fov, float aspect, float near, float far) {
75  float f = 1 / tan(fov / 2);
    float a = (far + near) / (near - far);

```



```

    float b = 2 * far * near / (near - far);
    float c = (a * (near + far) + 2 * b) / (near - far);
    float d = b * (near + far) - a * (near - far) / 2;
80    float o = 0;
    return glm::mat4(f/aspect,o,o,o, o,f,o,o, o,o,a,b, o,o,c,d);
}

void display(G *g, int frame) {
85    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    for (float x = -1; x <= 1; x += 1) {
        glm::mat4 p = glm::perspective(float(90.0f * twopi/360.0f), 16.0f / 9.0f, ↵
            ↵ 5.0f, 20.f);
        glm::mat4 t = glm::translate(glm::mat4(1.0f), glm::vec3(8.f * x, 0.0f, -8.f)↵
            ↵ );
        glm::mat4 r = glm::rotate(glm::mat4(1.0f), float(frame * twopi/360), glm::↵
            ↵ vec3(1.f, 1.f, 0.f));
90    glm::mat4 s = glm::scale(glm::mat4(1.0f), glm::vec3(2.0f));
        glm::mat4 m = p * t * r * s;
        glUniformMatrix4fv(g->mvp, 1, GL_FALSE, &m[0][0]);
        glDrawArrays(GL_TRIANGLES, 0, 36);
    }
95 }

void capture(unsigned char *b) {
    glReadPixels(0, 0, width, height, GL_RGB, GL_UNSIGNED_BYTE, b);
    printf("P6\n%d %d\n255\n", width, height);
100    fflush(stdout);
    fwrite(b, width * height * 3, 1, stdout);
    fflush(stdout);
}

105 int main(int argc, char **argv) {
    (void) argc;
    (void) argv;
    G g;
    unsigned char b[width * height * 3];
110    glfwInit();
    glfwWindowHint(GLFW_CLIENT_API, GLFW_OPENGL_API);
    glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, 3);
    glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, 3);
    glfwWindowHint(GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE);
115    glfwWindowHint(GLFW_OPENGL_FORWARD_COMPAT, GL_TRUE);
    glfwWindowHint(GLFW_RESIZABLE, GL_FALSE);
    GLFWwindow *window = glfwCreateWindow(width, height, "byzantine", 0, 0);
    glfwMakeContextCurrent(window);
    glewExperimental = GL_TRUE;
120    glewInit();
    glGetError();
    init(&g);
    int frame = 0;
    while (! glfwWindowShouldClose(window)) {
125        display(&g, frame++);
        glfwSwapBuffers(window);
        glfwPollEvents();
        int e = glGetError();
        if (e) fprintf(stderr, "%d\n", e);
130    //    capture(&b[0]);

```

```
//    if (frame == 360) { exit(0); }
    }
    return 0;
}
```

49 sketches/byzantine/.gitignore

byzantine

50 sketches/byzantine/Makefile

```
byzantine: byzantine.cc
    g++ -std=c++11 -Wall -pedantic -Wextra -O3 -o byzantine byzantine.cc -\
    ↪ lGL -lGLEW -lglfw
```

51 sketches/colour/colour.c

```
#include <stdio.h>

int header(FILE *f, int *w, int *h, int *d) {
    if (2 == fscanf(f, "P6\n%d %d\n255", w, h)) {
5        if ('\n' == getc(f)) {
            *d = 255;
            return 1;
        }
    }
10    return 0;
}

int main(int argc, char **argv) {
    int w = 0, h = 0, d = 0;
15    while (header(stdin, &w, &h, &d)) {
        if (d != 255) {
            break;
        }
        fprintf(stdout, "P6\n%d %d\n%d\n", w, h, 255);
20        for (int j = 0; j < h; ++j) {
            for (int i = 0; i < w; ++i) {
                int r = getc(stdin);
                int g = getc(stdin);
                int b = getc(stdin);
25                b = r >> 1;
                putc(r, stdout);
                putc(g, stdout);
                putc(b, stdout);
            }
30        }
    }
    return 0;
}
```

52 sketches/colour/Makefile

```
colour: colour.c
    gcc -std=c99 -Wall -pedantic -Wextra -O3 -o colour colour.c
```

53 sketches/craterscape/craterscape.cpp

```

/* OpenGL example code - Tessellation
 *
 * This example shows the usage of tessellation for terrain LOD.
 * The terrain is given as a texture of 3d samples (generalized
5 * heightfield) and gets rendered without use of a vbo/vao. Instead
 * sample coordinates are generated from InstanceID and VertexID.
 * Tessellation is used to dynamically change the amount of vertices
 * depending on distance from the viewer.
 * This example requires at least OpenGL 4.0
10 *
 * Autor: Jakob Progsch
 */

/* index
15 * line 122: vertex shader generates vertices from InstanceID/VertexID
 * line 139: tessellation control shader
 * line 163: tessellation evaluation shader
 * line 182: fragment shader with simple phong lighting
 * line 202: shader compilation
20 * line 274: terrain generation
 * line 330: input handling
 * line 415: draw call
 */

25 #include <GL/glew.h>
#include <GLFW/glfw3.h>

#include <glm/glm.hpp>
#include <glm/gtc/matrix_transform.hpp>
30 #include <glm/gtc/type_ptr.hpp>

#include <iostream>
#include <string>
#include <vector>
35
const double pi = 3.141592653589793;

bool running;

40 // window close callback function
int closedWindow()
{
    running = false;
    return GL_TRUE;
45 }

// helper to check and display for shader compiler errors
bool check_shader_compile_status(GLuint obj)
{
50     GLint status;
    glGetShaderiv(obj, GL_COMPILE_STATUS, &status);
    if(status == GL_FALSE)
    {
        GLint length;
55     glGetShaderiv(obj, GL_INFO_LOG_LENGTH, &length);

```

```

        std::vector<char> log(length);
        glGetShaderInfoLog(obj, length, &length, &log[0]);
        std::cerr << &log[0];
        return false;
60     }
        return true;
    }

    // helper to check and display for shader linker error
65    bool check_program_link_status(GLuint obj)
    {
        GLint status;
        glGetProgramiv(obj, GL_LINK_STATUS, &status);
        if(status == GL_FALSE)
70     {
            GLint length;
            glGetProgramiv(obj, GL_INFO_LOG_LENGTH, &length);
            std::vector<char> log(length);
            glGetProgramInfoLog(obj, length, &length, &log[0]);
75     std::cerr << &log[0];
            return false;
        }
        return true;
    }
80

    const GLfloat subdivision_data[8][4][4] =
    { // 0
        { { 3, 0, 1, 2 }
          , { 0, 4, 5, 6 }
85     , { 5, 1, 3, 5 }
          , { 7, 6, 1, 0 }
        }
        // 1
        , { { 3, 3, 0, 4 }
          , { 1, 0, 4, 3 }
90     , { 6, 4, 3, 3 }
          , { 4, 0, 1, 0 }
        }
        // 2
95     , { { 0, 4, 5, 4 }
          , { 2, 0, 1, 3 }
          , { 5, 4, 5, 1 }
          , { 7, 1, 0, 6 }
        }
100    // 3
        , { { 0, 1, 3, 2 }
          , { 2, 5, 1, 2 }
          , { 5, 7, 6, 4 }
          , { 7, 7, 4, 0 }
105    }
        // 4
        , { { 0, 4, 2, 2 }
          , { 4, 0, 4, 2 }
          , { 7, 6, 7, 4 }
110    , { 7, 4, 3, 0 }
        }
        // 5

```

```

    , { { 0, 6, 4, 2 }
      , { 5, 4, 3, 5 }
115    , { 0, 7, 1, 0 }
      , { 4, 3, 5, 6 }
    }
    // 6
    , { { 3, 5, 6, 4 }
      , { 7, 1, 4, 7 }
120    , { 1, 2, 3, 0 }
      , { 4, 2, 0, 6 }
    }
    // 7
125    , { { 0, 1, 3, 2 }
      , { 6, 4, 0, 6 }
      , { 4, 3, 5, 6 }
      , { 7, 7, 7, 6 }
    }
130 };

glm::vec3 hsv2rgb(glm::vec3 hsv)
{
    float h = hsv.x;
135    float s = hsv.y;
    float v = hsv.z;
    float i, f, p, q, t, r, g, b;
    int ii;
    if (s == 0.0) {
140        // Ignore hue
        r = v;
        g = v;
        b = v;
    } else {
145        /* Apply physiological mapping: red, yellow, green and blue should
           be equidistant. */
        h = h - floor(h);
        h = h * 6.0;
        i = floor(h); ii = (int) i;
150        f = h - i;
        p = v*(1.0 - s); // The "low-flat" curve
        q = v*(1.0 - (s*f)); // The "falling" curve
        t = v*(1.0 - (s*(1.0 - f))); // The "rising" curve
        switch(ii) {
155            case 0: // red point
                r = v; g = t; b = p; // RED max, GRN rising, BLU min
                break;
                case 1: // yellow point
                    r = q; g = v; b = p; // RED falling, GRN max, BLU min
160                break;
                case 2: // green point
                    r = p; g = v; b = t; // RED min, GRN max, BLU rising
                    break;
                case 3: // cyan point
165                r = p; g = q; b = v; // RED min, GRN falling, BLU max
                    break;
                case 4: // blue point
                    r = t; g = p; b = v; // RED rising, GRN min, BLU max
                    break;

```

```

170         case 5: // magenta point
            default:
                r = v; g = p; b = q; // RED max, GRN min, BLU falling
                break;
        }
175     }
    return glm::vec3(r, g, b);
}

void craters(std::vector<glm::vec4> &colourData, std::vector<GLfloat> &displacementData, int width, int height, int layers) {
180     int subdivide = 4;
    double diameter = 256;
    double depth = 1.0 / 16;
    double density = 4;
    int count = density * subdivide * width * height / (diameter * diameter);
185     double rim = 0.25;
    double bowl = 1 - rim;
    for (int k = 0; k < count; ++k) {
        // crater area is inversely proportional to frequency
        // this is hardcoded to 4x4 subdivision
190         double p = (rand() + 1.0) / RANDMAX; // 0 < p <= 1
            p = exp2(fmod(-log2(p), 1.0)); // 1 <= p < 2
            p = 1.0 / (p * p); // 1/4 < p <= 1
        GLfloat hue = rand() / (double) RANDMAX;
        glm::vec4 colour = glm::vec4(hsv2rgb(GLfloat(p) * glm::vec3(hue, 1.0f, 1.0f)
195             ↵ ), GLfloat(p));
        double r = 0.5 * diameter * p;
        double x = rand() / (double) RANDMAX * width;
        double y = rand() / (double) RANDMAX * height;
        int t0 = ((int) (rand() / (double) RANDMAX * layers)) % layers;
        double r2 = r * r;
200         // reject craters that cover edges
        if (x <= r || y <= r || width - x <= r || height - y < r) {
            continue;
        }
        // rasterize crater
205         int lj = floor(y - r);
        int hj = ceil(y + r);
        int li = floor(x - r);
        int hi = ceil(x + r);
        #pragma omp parallel for
210         for (int j = lj; j <= hj; ++j) {
            for (int i = li; i <= hi; ++i) {
                double dy = j - y;
                double dx = i - x;
                double dy2 = dy * dy;
215                 double dx2 = dx * dx;
                double d2 = dx2 + dy2;
                if (d2 < r2) {
                    double z = sqrt(d2) / r * (bowl + 4.0 * rim / 3.0);
                    double h = 0;
220                     if (z < bowl) {
                        h = tanh(2 * (pow(z / bowl, 2) - 1)) / 2; // d/dz @ bowl = 2 / bowl
                        h += (2 / bowl) / (3 * pi / (2 * rim));
                    } else if (z < bowl + rim) {
                        h = sin(1.5 * pi * (z - bowl) / rim) + 1; // d/dz @ bowl = 3 pi / (2

```

```

    ↪ rim)
225     h *= (2 / bowl) / (3 * pi / (2 * rim));
    }
    h *= -depth * p;
    assert(0 <= j && j < height);
    assert(0 <= i && i < width);
230 //    if (h < displacementData[i + j * width + t0 * width * height]) {
        displacementData[i + j * width + t0 * width * height] += h;
        float edge = fmin(fmax(0.01 / pow((z - bowl - -1.0/3.0 * rim) / (rim ↪
            ↪ / 6.0), 8.0) - 1.0, 0.0), 1.0)
            + fmin(fmax(0.01 / pow((z - bowl - 1.0/3.0 * rim) / (rim ↪
            ↪ / 6.0), 8.0) - 1.0, 0.0), 1.0)
            + fmin(fmax(0.01 / pow((z - bowl - 3.0/3.0 * rim) / (rim ↪
            ↪ / 6.0), 8.0) - 1.0, 0.0), 1.0);
235     colourData[i + j * width + t0 * width * height] += colour * glm::↪
        ↪ vec4(edge, edge, edge, 1.0);
    //    }
        }
    }
240 }
}

int main()
{
245     srand(time(0));
    int width = 1280;
    int height = 720;

    if (glfwInit() == GL_FALSE)
250     {
        std::cerr << "failed to init GLFW" << std::endl;
        return 1;
    }

255     // we need a 4.0 profile this time
    glfwWindowHint(GLFW_OPENGL_PROFILE, GLFW_OPENGL_CORE_PROFILE);
    glfwWindowHint(GLFW_CONTEXT_VERSION_MAJOR, 4);
    glfwWindowHint(GLFW_CONTEXT_VERSION_MINOR, 0);

260     // create a window
    GLFWwindow *window = 0;
    if (!(window = glfwCreateWindow(width, height, "craterscape", 0, 0)))
    {
        std::cerr << "failed to open window" << std::endl;
265         glfwTerminate();
        return 1;
    }

    glfwMakeContextCurrent(window);
270     glewExperimental = GL_TRUE;
    glewInit();
    glGetError();

    glfwSwapInterval(1);
275

    // this time we disable the mouse cursor since we want differential

```

```

// mouse input
// glfwDisable(GLFW_MOUSE_CURSOR);

280     GLuint vao;
        glGenVertexArrays(1, &vao);
        glBindVertexArray(vao);

// shader source code
285     std::string vertex_source =
        "#version 400\n"
        "uniform uint width;\n"
        "uniform uint height;\n"
        "out vec4 tposition;\n"
290     "const vec2 quad_offsets[6] = vec2[](\n"
        "    vec2(0,0),vec2(1,0),vec2(1,1),\n"
        "    vec2(0,0),vec2(1,1),vec2(0,1)\n"
        ");\n"
        "void main() {\n"
295     "    vec2 base = vec2(gl_InstanceID*width, gl_InstanceID/width);\n"
        "    vec2 offset = quad_offsets[gl_VertexID];\n"
        "    vec2 pos = (base + offset)/vec2(width, height);\n"
        "    tposition = vec4(pos,0,1);\n"
        "}\n";

300     std::string tess_control_source =
        "#version 400\n"
        "uniform vec3 ViewPosition;\n"
        "uniform float tess_scale;\n"
305     "layout(vertices = 3) out;\n"
        "in vec4 tposition[];\n"
        "out vec4 tcposition[];\n"
        "void main()\n"
        "{\n"
310     "    tcposition[gl_InvocationID] = tposition[gl_InvocationID];\n"
        "    if(gl_InvocationID == 0) {\n"
        "        vec3 terrainpos = ViewPosition;\n"
        "        terrainpos.z -= clamp(terrainpos.z, -1.0, 1.0);\n"
        "        vec4 center = (tcposition[1]+tcposition[2])/2.0;\n"
315     "        gl_TessLevelOuter[0] = min(16.0, 1+tess_scale*0.5/distance(
            ↪ center.xyz, terrainpos));\n"
        "        center = (tcposition[2]+tcposition[0])/2.0;\n"
        "        gl_TessLevelOuter[1] = min(16.0, 1+tess_scale*0.5/distance(
            ↪ center.xyz, terrainpos));\n"
        "        center = (tcposition[0]+tcposition[1])/2.0;\n"
        "        gl_TessLevelOuter[2] = min(16.0, 1+tess_scale*0.5/distance(
            ↪ center.xyz, terrainpos));\n"
320     "        center = (tcposition[0]+tcposition[1]+tcposition[2])/3.0;\n"
        "        gl_TessLevelInner[0] = min(16.0, 1+tess_scale*0.7/distance(
            ↪ center.xyz, terrainpos));\n"
        "    }\n"
        "}\n";

325     std::string tess_eval_source =
        "#version 400\n"
        "uniform vec2 scroll;\n"
        "uniform mat4 ViewProjection;\n"
        "uniform sampler2DArray displacement;\n"

```



```

330     "uniform sampler2DArray subdivision;\n"
    "layout(triangles , equal_spacing , cw) in;\n"
    "in vec4 tposition [];\n"
    "out vec3 tecoord0;\n"
    "out vec4 teposition;\n"
335     "void main()\n"
    "{\n"
    "    vec4 teposition0 = gl_TessCoord.x * tposition[0]\n"
    "        + gl_TessCoord.y * tposition[1]\n"
    "        + gl_TessCoord.z * tposition[2];\n"
340     "    vec3 tecoord = vec3(teposition0.xy + scroll , 0.0);\n"
    "    tecoord0 = tecoord;\n"
    "    vec3 offset = vec3(tecoord.xy , 0.0);\n"
    "    float scale = 1.0;\n"
    "    float factor = 4.0;\n"
345     "    for (int i = 0; i < 2; ++i) {\n"
    "        offset.z += scale * texture(displacement , tecoord).x;\n"
    "        tecoord.z = texture(subdivision , tecoord).x;\n"
    "        tecoord.xy *= factor;\n"
    "        scale /= factor;\n"
350     "    }\n"
    "    teposition0.xyz = offset;\n"
    "    teposition0.xy -= scroll;\n"
    "    teposition = teposition0;\n"
    "    gl_Position = ViewProjection*teposition0;\n"
355     "}\n";

std::string fragment_source =
    "#version 400\n"
    "uniform vec3 ViewPosition;\n"
360     "uniform vec3 LightPosition;\n"
    "uniform sampler2DArray displacement;\n"
    "uniform sampler2DArray colour;\n"
    "uniform sampler2DArray subdivision;\n"
    "in vec4 tposition;\n"
365     "in vec3 tecoord0;\n"
    "layout(location = 0) out vec4 FragColor;\n"
    "void main() {\n"
    "    vec4 c = vec4(0.0,0.0,0.0,0.1);\n"
    "    vec3 tecoord = tecoord0;\n"
370     "    vec3 x = vec3(tecoord.xy , 0.0);\n"
    "    float scale = 1.0;\n"
    "    float factor = 4.0;\n"
    "    for (int i = 0; i < 2; ++i) {\n"
    "        float z = scale * texture(displacement , tecoord).x;\n"
375     "        x.z += z;\n"
    "        c += scale * texture(colour , tecoord);\n"
    "        t0.z += scale * textureOffset(displacement , tecoord , ivec2\
//      ↪ (1,0)).x;\n"
    "        t1.z += scale * textureOffset(displacement , tecoord , ivec2\
//      ↪ (0,1)).x;\n"
    "        tecoord.z = texture(subdivision , tecoord).x;\n"
380     "        tecoord.xy *= factor;\n"
    "        scale /= factor;\n"
    "    }\n"
    "    vec3 normal = (gl_FrontFacing?1:-1)*normalize(cross(x-t0 , x-t1));\n\
//      ↪ n"

```

```

385 //      "    vec3 light = -normalize(LightPosition-teposition.xyz);\n"
//      "    vec3 reflected = reflect(normalize(ViewPosition-teposition.xyz), ↗
    ↘ normal);\n"
//      "    float ambient = 0.1 / distance(LightPosition, teposition.xyz);\n"
//      "    float diffuse = ambient * max(0,dot(normal, light));\n"
//      "    float specular = ambient * pow(max(0,dot(reflected, light)), 64);\n↗
    ↘ n"
    "    c.rgb /= c.a;"
390 //      "    c.rgb *= 2.0 * (ambient + diffuse + specular);\n"
//      "    c.rgb -= min(min(c.r, c.g), c.b) * 0.5;"
//      "    c.rgb /= max(max(c.r, c.g), c.b);"
    "    float fog = clamp(1.0 - 2.0 * distance(ViewPosition, teposition.xyz)↗
    ↘ , 0.0, 1.0);\n"
//      "    float fog = 1.0;\n"
395 "    FragColor = vec4(mix(vec3(0.0), 2.0 * c.rgb, fog * fog), 1);\n"
    "}\n";

// program and shader handles
GLuint shader_program, vertex_shader, tess_control_shader, tess_eval_shader, ↗
    ↘ fragment_shader;

400

// we need these to properly pass the strings
const char *source;
int length;

405 // create and compiler vertex shader
vertex_shader = glCreateShader(GL_VERTEX_SHADER);
source = vertex_source.c_str();
length = vertex_source.size();
glShaderSource(vertex_shader, 1, &source, &length);
410 glCompileShader(vertex_shader);
if(!check_shader_compile_status(vertex_shader))
{
    return 1;
}

415 // create and compiler tessellation control shader
tess_control_shader = glCreateShader(GL_TESS_CONTROL_SHADER);
source = tess_control_source.c_str();
length = tess_control_source.size();
420 glShaderSource(tess_control_shader, 1, &source, &length);
glCompileShader(tess_control_shader);
if(!check_shader_compile_status(tess_control_shader))
{
    return 1;
425 }

// create and compiler tessellation evaluation shader
tess_eval_shader = glCreateShader(GL_TESS_EVALUATION_SHADER);
source = tess_eval_source.c_str();
430 length = tess_eval_source.size();
glShaderSource(tess_eval_shader, 1, &source, &length);
glCompileShader(tess_eval_shader);
if(!check_shader_compile_status(tess_eval_shader))
{
435     return 1;
}

```

```

// create and compiler fragment shader
fragment_shader = glCreateShader(GL_FRAGMENT_SHADER);
440 source = fragment_source.c_str();
length = fragment_source.size();
glShaderSource(fragment_shader, 1, &source, &length);
glCompileShader(fragment_shader);
if (!check_shader_compile_status(fragment_shader))
445 {
    return 1;
}

// create program
450 shader_program = glCreateProgram();

// attach shaders
glAttachShader(shader_program, vertex_shader);
glAttachShader(shader_program, tess_control_shader);
455 glAttachShader(shader_program, tess_eval_shader);
glAttachShader(shader_program, fragment_shader);

// link the program and check for errors
glLinkProgram(shader_program);
460 check_program_link_status(shader_program);

GLint width_Location = glGetUniformLocation(shader_program, "width");
GLint height_Location = glGetUniformLocation(shader_program, "height");
GLint ViewProjection_Location = glGetUniformLocation(shader_program, "view_
↳ ViewProjection");
465 GLint ViewPosition_Location = glGetUniformLocation(shader_program, "view_
↳ ViewPosition");
GLint LightPosition_Location = glGetUniformLocation(shader_program, "light_
↳ LightPosition");
GLint displacement_Location = glGetUniformLocation(shader_program, "dis_
↳ displacement");
GLint colour_Location = glGetUniformLocation(shader_program, "colour");
GLint subdivision_Location = glGetUniformLocation(shader_program, "sub_
↳ subdivision");
470 GLint tess_scale_Location = glGetUniformLocation(shader_program, "tess_scale_
↳ ");
GLint scroll_Location = glGetUniformLocation(shader_program, "scroll");

int terrainwidth = 1024, terrainheight = 1024;
475 std::vector<GLfloat> displacementData(terrainwidth*terrainheight*8);
std::vector<glm::vec4> colourData(terrainwidth*terrainheight*8);
craters(colourData, displacementData, terrainwidth, terrainheight, 8);

// texture handle
480 GLuint displacement;

// generate texture
glGenTextures(1, &displacement);

485 // bind the texture
glBindTexture(GL_TEXTURE_2D_ARRAY, displacement);

```

```

// set texture parameters
glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_MIN_FILTER, ↵
    ↵ GL_LINEAR_MIPMAP_LINEAR);
490 glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_WRAP_T, GL_REPEAT);

// set texture content
495 glTexImage3D(GL_TEXTURE_2D_ARRAY, 0, GL_R32F, terrainwidth, terrainheight, ↵
    ↵ 8, 0, GL_RED, GL_FLOAT, &displacementData[0]);
glGenerateMipmap(GL_TEXTURE_2D_ARRAY);

// texture handle
GLuint colour;
500

// generate texture
glGenTextures(1, &colour);

// bind the texture
505 glBindTexture(GL_TEXTURE_2D_ARRAY, colour);

// set texture parameters
glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_MIN_FILTER, ↵
    ↵ GL_LINEAR_MIPMAP_LINEAR);
glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
510 glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_WRAP_T, GL_REPEAT);

// set texture content
glTexImage3D(GL_TEXTURE_2D_ARRAY, 0, GL_RGBA32F, terrainwidth, terrainheight ↵
    ↵ , 8, 0, GL_RGBA, GL_FLOAT, &colourData[0]);
515 glGenerateMipmap(GL_TEXTURE_2D_ARRAY);

GLuint subdivision;
glGenTextures(1, &subdivision);
glBindTexture(GL_TEXTURE_2D_ARRAY, subdivision);
520 glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_WRAP_S, GL_REPEAT);
glTexParameteri(GL_TEXTURE_2D_ARRAY, GL_TEXTURE_WRAP_T, GL_REPEAT);
glTexImage3D(GL_TEXTURE_2D_ARRAY, 0, GL_R32F, 4, 4, 8, 0, GL_RED, GL_FLOAT, ↵
    ↵ &subdivision_data[0][0][0]);
525

// camera position and orientation
glm::vec3 position(0.5, 0.5, -0.04);
glm::mat4 rotation = glm::mat4(1.0f);
rotation = glm::rotate(rotation, glm::radians(90.0f), glm::vec3(1.0f, 0.0f ↵
    ↵ , 0.0f));
530 glm::mat3 rotation3(rotation);
glm::vec3 up = glm::transpose(rotation3)*glm::vec3(0.0f, 1.0f, 0.0f);
GLfloat phi = (sqrt(5) + 1)/2;
rotation = glm::rotate(rotation, atan2f(phi, 1.0f), up);
// rotation = glm::rotate(rotation, 90.f, up);
535 // float t = glfwGetTime();
bool tessellation = true;
bool space_down = false;

```

```

    glEnable(GL_DEPTH_TEST);
540    // check for errors
    GLenum error = glGetError();
    if (error != GL_NO_ERROR)
    {
545        std::cerr << "preloop: " << gluErrorString(error);
        return 1;
    }
    // std::vector<unsigned char> buffer(width * height * 3);
    while (! glfwWindowShouldClose(window))
550    {
        // calculate timestep
        float dt = 0.04;
    /*
        float new_t = glfwGetTime();
555        float dt = new_t - t;
        t = new_t;
        std::cerr << dt << "\r";
    */

560    // find up, forward and right vector
    glm::mat3 rotation3(rotation);
    // glm::vec3 up = glm::transpose(rotation3)*glm::vec3(0.0f, 1.0f, 0.0f);
    glm::vec3 right = glm::transpose(rotation3)*glm::vec3(1.0f, 0.0f, 0.0f);
    glm::vec3 forward = glm::transpose(rotation3)*glm::vec3(0.0f, 0.0f, -1.0f)
        ↵ );
565    // apply mouse rotation
    // rotation = glm::rotate(rotation, 0.2f*mousediff.x, up);
    // rotation = glm::rotate(rotation, 0.2f*mousediff.y, right);
    // rotation = glm::rotate(rotation, 0.01f, up);
570    // roll
    if (glfwGetKey(window, 'Q'))
    {
        rotation = glm::rotate(rotation, glm::radians(180.0f*dt), forward);
575    }
    if (glfwGetKey(window, 'E'))
    {
        rotation = glm::rotate(rotation, -glm::radians(180.0f*dt), forward);
    }
580    float speed = 0.1f;
    // movement
    // if (glfwGetKey('W'))
    {
585        position += speed*dt*forward;
    }
    if (glfwGetKey(window, 'S'))
    {
        position -= speed*dt*forward;
590    }
    if (glfwGetKey(window, 'D'))
    {
        position += speed*dt*right;
    }

```

```

595     if (glfwGetKey(window, 'A'))
    {
        position -= speed*dt*right;
    }

600     // terminate on escape
    if (glfwGetKey(window, GLFW_KEY_ESCAPE))
    {
        break;
    }

605     // toggle tessellation
    if (glfwGetKey(window, GLFW_KEY_SPACE) && !space_down)
    {
        tessellation = !tessellation;
    }
610     space_down = glfwGetKey(window, GLFW_KEY_SPACE);

    // calculate ViewProjection matrix
    glm::vec2 scroll(round(position.x * 1024)/1024, round(position.y * 1024)↵
        ↵ /1024);
615     glm::mat4 Projection = glm::perspective(glm::radians(45.0f), float(width↵
        ↵ ) / height, 0.001f, 10.f);
    glm::mat4 View = glm::rotate(rotation, glm::radians(15.0f), right)*glm::↵
        ↵ translate(glm::mat4(1.0f), glm::vec3(-0.5 - position.x + scroll.x↵
        ↵ , -0.5 - position.y + scroll.y, -position.z));
    glm::mat4 ViewProjection = Projection*View;

    // clear first
620     glClearColor(0,0,0,1);
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

    glActiveTexture(GL_TEXTURE0);
    glBindTexture(GL_TEXTURE_2D_ARRAY, displacement);
625     glActiveTexture(GL_TEXTURE1);
    glBindTexture(GL_TEXTURE_2D_ARRAY, colour);
    glActiveTexture(GL_TEXTURE2);
    glBindTexture(GL_TEXTURE_2D_ARRAY, subdivision);

630     // use the shader program
    int gridsize = 64;
    glUseProgram(shader_program);
    glUniform1ui(width_Location, gridsize); // 64x64 base grid without ↵
        ↵ tessellation
    glUniform1ui(height_Location, gridsize);
635     glUniformMatrix4fv(ViewProjection_Location, 1, GL_FALSE, glm::value_ptr(↵
        ↵ ViewProjection));
    glUniform3f(ViewPosition_Location, 0.5, 0.5, position.z);
    glUniform3f(LightPosition_Location, 0.5, 0.5, -0.01);
    glUniform2f(scroll_Location, scroll.x, scroll.y);

640     if (tessellation)
        glUniform1f(tess_scale_Location, 5.0f);
    else
        glUniform1f(tess_scale_Location, 0.0f);

645     // set texture uniform

```

```

        glUniform1i(displacement_Location, 0);
        glUniform1i(colour_Location, 1);
        glUniform1i(subdivision_Location, 2);

650    // draw
        glPatchParameteri(GL_PATCH_VERTICES, 3);
        glDrawArraysInstanced(GL_PATCHES, 0, 6, gridsize * gridsize);

        // check for errors
655    GLenum error = glGetError();
        if (error != GL_NO_ERROR)
        {
            std::cerr << gluErrorString(error);
            break;
660    }

        // finally swap buffers
        glfwSwapBuffers(window);
        glfwPollEvents();

665    /*
        glReadPixels(0,0,width,height,GL_RGB,GL_UNSIGNED_BYTE,&buffer[0]);
        fprintf(stdout, "P6\n%d %d\n255\n", width, height);
        fwrite(&buffer[0], width * height * 3, 1, stdout);
        */
670    }
        // delete the created objects

        glDeleteVertexArrays(1, &vao);
        glDetachShader(shader_program, vertex_shader);
675    glDetachShader(shader_program, tess_control_shader);
        glDetachShader(shader_program, tess_eval_shader);
        glDetachShader(shader_program, fragment_shader);
        glDeleteShader(vertex_shader);
        glDeleteShader(tess_control_shader);
680    glDeleteShader(tess_eval_shader);
        glDeleteShader(fragment_shader);
        glDeleteProgram(shader_program);

        glfwDestroyWindow(window);
685    glfwTerminate();
        return 0;
}

```

54 sketches/craterscape/.gitignore

craterscape

55 sketches/craterscape/Makefile

```

craterscape: craterscape.cpp
    g++ -std=c++11 -Wall -pedantic -Wextra -lGL -lGLEW -lGLU -lglfw -o ʘ
    ʘ craterscape craterscape.cpp -O3 -fopenmp

```

56 sketches/craters/craters.c

```

#include <assert.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
5  #include <string.h>
#include <time.h>

#define PI 3.141592653589793
#define TILES 8
10 #define SUBDIVIDE 4
#define SIZE 1024
#define DIAMETER 256
#define DEPTH 0.25
#define DENSITY 0.666
15 #define COUNT ((DENSITY * SUBDIVIDE * SIZE * SIZE) / (DIAMETER * DIAMETER))
#define RIM (1.5 * PI * 8.0)
#define BOWL (1.0 - 1.5 * PI / RIM)

const unsigned char neighbours[TILES][4] =
20  // N E S W
  { { 0, 1, 2, 3 }
    , { 0, 3, 2, 1 }
    , { 2, 3, 2, 3 }
    , { 0, 3, 0, 3 }
25  , { 2, 3, 0, 1 }
    , { 2, 1, 0, 3 }
    , { 2, 1, 2, 1 }
    , { 0, 1, 0, 1 }
  };
30

const unsigned char subdivision[TILES][SUBDIVIDE][SUBDIVIDE] =
  { // 0
    { { 3, 0, 1, 2 }
      , { 0, 4, 5, 6 }
35    , { 5, 1, 3, 5 }
      , { 7, 6, 1, 0 }
    }
    // 1
    , { { 3, 3, 0, 4 }
40    , { 1, 0, 4, 3 }
      , { 6, 4, 3, 3 }
      , { 4, 0, 1, 0 }
    }
    // 2
45    , { { 0, 4, 5, 4 }
      , { 2, 0, 1, 3 }
      , { 5, 4, 5, 1 }
      , { 7, 1, 0, 6 }
    }
50    // 3
    , { { 0, 1, 3, 2 }
      , { 2, 5, 1, 2 }
      , { 5, 7, 6, 4 }
      , { 7, 7, 4, 0 }
55    }
    // 4
    , { { 0, 4, 2, 2 }

```



```

        , { 4, 0, 4, 2 }
        , { 7, 6, 7, 4 }
60    , { 7, 4, 3, 0 }
        }
        // 5
        , { { 0, 6, 4, 2 }
        , { 5, 4, 3, 5 }
65    , { 0, 7, 1, 0 }
        , { 4, 3, 5, 6 }
        }
        // 6
        , { { 3, 5, 6, 4 }
70    , { 7, 1, 4, 7 }
        , { 1, 2, 3, 0 }
        , { 4, 2, 0, 6 }
        }
        // 7
75    , { { 0, 1, 3, 2 }
        , { 6, 4, 0, 6 }
        , { 4, 3, 5, 6 }
        , { 7, 7, 7, 6 }
        }
80    };

float height[TILES][SIZE][SIZE];

double accum(int t, double y, double x, int d) {
85    if (d > 0) {
        int j = floor(y);
        int i = floor(x);
        int tt = subdivision[t][(j * SUBDIVIDE) / SIZE][(i * SUBDIVIDE) / SIZE];
        return height[t][j][i] + accum(tt, fmod(y * SUBDIVIDE, SIZE), fmod(x *
            SUBDIVIDE, SIZE), d - 1) / SUBDIVIDE;
90    } else {
        return 0;
    }
}

95 unsigned char pgm[SIZE][SIZE];

int main(int argc, char **argv) {
    srand(time(0));
100    memset(height, 0, TILES * SIZE * SIZE * sizeof(float));
    for (int k = 0; k < COUNT; ++k) {
        // crater area is inversely proportional to frequency
        // this is hardcoded to 4x4 subdivision
        double p = (rand() + 1.0) / RANDMAX; // 0 < p <= 1
105        p = exp2(fmod(-log2(p), 1.0)); // 1 <= p < 2
        p = 1.0 / (p * p); // 1/4 < p <= 1
        double r = 0.5 * DIAMETER * p;
        double x = rand() / (double) RANDMAX * SIZE;
        double y = rand() / (double) RANDMAX * SIZE;
110    int t0 = ((int) (rand() / (double) RANDMAX * TILES)) % TILES;
        double r2 = r * r;
        // reject craters that cover edges
        if (x <= r || y <= r || SIZE - x <= r || SIZE - y < r) {

```

```

        continue;
115    }
    // rasterize crater
    int lj = floor(y - r);
    int hj = ceil(y + r);
    int li = floor(x - r);
120    int hi = ceil(x + r);
    #pragma omp parallel for
    for (int j = lj; j <= hj; ++j) {
        for (int i = li; i <= hi; ++i) {
            double dy = j - y;
            double dx = i - x;
125            double dy2 = dy * dy;
            double dx2 = dx * dx;
            double d2 = dx2 + dy2;
            if (d2 < r2) {
130                double z = sqrt(d2) / (BOWL * r);
                double h = 0;
                if (z < 1) {
                    h = tanh(pow(z, 2) - 1) / 2 + 1 / RIM;
                } else {
135                    h = (sin(RIM * (z - 1)) + 1) / RIM;
                }
                h *= DEPTH * r;
                assert(0 <= j && j < SIZE);
                assert(0 <= i && i < SIZE);
140                height[t0][j][i] += h;
            }
        }
    }
}

145 // rescale height to output pgm
float mi = 1.0f / 0.0f;
float ma = -1.0f / 0.0f;
for (int t = 0; t < TILES; ++t) {
    for (int j = 0; j < SIZE; ++j) {
150        for (int i = 0; i < SIZE; ++i) {
            mi = fminf(mi, height[t][j][i]);
            ma = fmaxf(ma, height[t][j][i]);
        }
    }
155 }
float k = 255.0 / (ma - mi);
printf("min: %f\nzero: %f\nmax: %f\n", mi, k * (0 - mi), ma);
char filename[] = "crater-X.pgm";
for (int t = 0; t < TILES; ++t) {
160     filename[7] = '0' + t;
    FILE *f = fopen(filename, "wb");
    fprintf(f, "P5\n%d %d\n255\n", SIZE, SIZE);
    #pragma omp parallel for
    for (int j = 0; j < SIZE; ++j) {
165        for (int i = 0; i < SIZE; ++i) {
            pgm[j][i] = fminf(fmaxf(k * (height[t][j][i] - mi), 0), 255);
        }
    }
    fwrite(&pgm[0][0], SIZE * SIZE, 1, f);
170    fclose(f);
}

```

```

    }
    // render an example using multiple levels
    // sum_{i=0}^inf 1/4^i = 4 / 3
    mi *= 4.0 / 3.0;
175    ma *= 4.0 / 3.0;
    k = 255.0 / (ma - mi);
    FILE *f = fopen("craters.pgm", "wb");
    fprintf(f, "P5\n%d %d\n255\n", SIZE, SIZE);
    #pragma omp parallel for
180    for (int j = 0; j < SIZE; ++j) {
        for (int i = 0; i < SIZE; ++i) {
            double h = 0;
            for (int jj = 0; jj < 4; ++jj) {
                for (int ii = 0; ii < 4; ++ii) {
185                    h += accum(0, j + jj / 4.0, i + ii / 4.0, 16);
                }
            }
            h /= 16.0;
            pgm[j][i] = fminf(fmaxf(k * (h - mi), 0), 255);
190        }
    }
    fwrite(&pgm[0][0], SIZE * SIZE, 1, f);
    fclose(f);
    return 0;
195 }

```

57 sketches/craters/.gitignore

```

craters
*.pgm

```

58 sketches/craters/Makefile

```

craters: craters.c
    gcc -std=c99 -Wall -pedantic -Wextra -O3 -o craters craters.c -lm -x
    ↪ fopenmp

```

59 sketches/dolly/3D.frag

```

#donotrun

/*
Simple 3D setup with anti-alias and DOF.
5
Implement this function in fragments that include this file:
vec3 color(vec3 cameraPos, vec3 direction);

*/
10
#buffer RGBA32F
#buffershader "BufferShader.frag"

#info Simple 3D Setup
15 #camera 3D

#vertex

```

```

#group Camera
20
    uniform float time;
    vec3 Eye = vec3(0.0, 0.187, 5.0 - 5.0 * cos(6.283185307179586 * time / 30.0));
    vec3 Target = vec3(0.0, 0.187, 10.5);
    vec3 Up = vec3(0.0, 1.0, 0.0);
25    float FOV = atan(1.0, Target.z - Eye.z);

    varying vec3 from;
    uniform vec2 pixelSize;
    varying vec2 coord;
30    varying vec2 viewCoord;
    varying vec2 viewCoord2;
    varying vec3 dir;
    varying vec3 Dir;
    varying vec3 UpOrtho;
35    varying vec3 Right;
    uniform int backbufferCounter;
    varying vec2 PixelScale;

#ifdef providesInit
40    void init(); // forward declare
#else
    void init() {}
#endif

45    vec2 rand(vec2 co){
        // implementation found at: lumina.sourceforge.net/Tutorials/Noise.html
        return
        vec2(fract(sin(dot(co.xy ,vec2(12.9898,78.233))) * 43758.5453),
            fract(cos(dot(co.xy ,vec2(4.898,7.23))) * 23421.631));
50    }

void main(void)
55    {
        gl_Position = gl_Vertex;
        coord = (gl_ProjectionMatrix*gl_Vertex).xy;
        coord.x*= pixelSize.y/pixelSize.x;

60        // we will only use gl_ProjectionMatrix to scale and translate, so the ↯
        ↯ following should be OK.
        PixelScale =vec2(pixelSize.x*gl_ProjectionMatrix[0][0], pixelSize.y*↯
        ↯ gl_ProjectionMatrix[1][1]);
        viewCoord = gl_Vertex.xy;
        viewCoord2= (gl_ProjectionMatrix*gl_Vertex).xy;

65        from = Eye;
        Dir = normalize(Target-Eye);
        UpOrtho = normalize( Up-dot(Dir,Up)*Dir );
        Right = normalize( cross(Dir,UpOrtho));
        coord*=FOV;
70        init();
    }
#endvertex

```

```

#group Camera
75  uniform bool EquiRectangular; checkbox[ false]

#group Raytracer

80  #define PI  3.14159265358979323846264

    // Camera position and target.
    varying vec3 from;
    varying vec3 dir;
85  varying vec3 dirDx;
    varying vec3 dirDy;
    varying vec2 coord;
    varying float zoom;

90  uniform int backbufferCounter;
    uniform sampler2D backbuffer;
    varying vec2 viewCoord;
    varying vec2 viewCoord2;
    varying vec3 Dir;
95  varying vec3 UpOrtho;
    varying vec3 Right;
    uniform float time;
    vec3 Eye = vec3(0.0, 0.187, 5.0 - 5.0 * cos(6.283185307179586 * time / 30.0));
    vec3 Target = vec3(0.0, 0.187, 10.5);
100 float FocalPlane = Target.z - Eye.z;
    uniform float Aperture; slider[0,0.00,0.2]

    vec2 rand2(vec2 co){
        // implementation found at: lumina.sourceforge.net/Tutorials/Noise.html
105         return
            vec2(fract(sin(dot(co.xy ,vec2(12.9898,78.233))) * 43758.5453),
                fract(cos(dot(co.xy ,vec2(4.898,7.23))) * 23421.631));
    }

110  vec2 uniformDisc(vec2 co) {
        vec2 r = rand2(co);
        return sqrt(r.y)*vec2(cos(r.x*6.28),sin(r.x*6.28));
    }

115  #ifndef providesInit
    void init(); // forward declare
    #else
    void init() {}
    #endif
120  //out vec4 gl_FragColor;
    #group Post
    uniform float Gamma; slider[0.0,1.0,5.0]
    // 1: Linear, 2: Expontial, 3: Filmic, 4: Reinhart; 5: Syntopia
    uniform int ToneMapping; slider[1,1,5]
125  uniform float Exposure; slider[0.0,1.0,3.0]
    uniform float Brightness; slider[0.0,1.0,5.0];
    uniform float Contrast; slider[0.0,1.0,5.0];
    uniform float Saturation; slider[0.0,1.0,5.0];
    uniform float GaussianWeight; slider[0.0,1.0,10.0];

```

```

130    uniform float AntiAliasScale; slider [0,2,10]

    varying vec2 PixelScale;
    float FOV = atan(1.0, Target.z - Eye.z);
135    // implement this;
    vec3 color(vec3 cameraPos, vec3 direction);

    // Given a camera pointing in 'dir' with an orthogonal 'up' and 'right' vector
140    // and a point, coord, in screen coordinates from (-1,-1) to (1,1),
    // a ray tracer direction is returned
    vec3 equiRectangularDirection(vec2 coord, vec3 dir, vec3 up, vec3 right) {
        vec2 r = vec2(coord.x, (1.0 - coord.y) * 0.5) * PI;
        return cos(r.x) * sin(r.y) * dir +
145        sin(r.x) * sin(r.y) * right +
        cos(r.y) * up;
    }

150    void main() {
        init();
        vec3 hitNormal = vec3(0.0);
        vec3 hit;

155        // A Thin Lens camera model with Depth-Of-Field

        // We want to sample a circular diaphragm
        // Notice: this is not sampled with uniform density
        vec2 r = Aperture * uniformDisc(viewCoord * (float(backbufferCounter) + 1.0));
160        vec2 disc = uniformDisc(coord * float(1 + backbufferCounter)); // subsample ↯
        ↪ jitter
        vec2 jitteredCoord = coord + AntiAliasScale * PixelScale * FOV * disc;

        // Direction from Lens position to point on FocalPlane
        vec3 lensOffset = r.x * Right + r.y * UpOrtho;
165        // vec3 rayDir = normalize (Dir + jitteredCoord.x * Right + jitteredCoord.y * ↯
        ↪ UpOrtho) * FocalPlane - (lensOffset);
        vec3 rayDir = (Dir + jitteredCoord.x * Right + jitteredCoord.y * UpOrtho) * ↯
        ↪ FocalPlane - (lensOffset);
        rayDir = normalize(rayDir);

        if (EquiRectangular) {
170            rayDir = equiRectangularDirection(viewCoord2, Dir, UpOrtho, ↯
            ↪ Right);
        }

        vec3 c = color(from + lensOffset, rayDir);

175        // Accumulate
        vec4 prev = texture2D(backbuffer, (viewCoord + vec2(1.0)) / 2.0);
        float w = 1.0 - length(disc);
        if (GaussianWeight > 0.) {
            w = exp(-dot(disc, disc) / GaussianWeight) - exp(-1. / GaussianWeight);
180        }
        gl_FragColor = (prev + vec4(c * w, w));
    }

```

60 sketches/dolly/BufferShader.frag

```

#donotrun
// This is a simple shader for rendering images
// from an accumulated buffer.

5  #vertex

    varying vec2 coord;

    void main(void)
10  {
        gl_Position = gl_Vertex;
        coord = (gl_ProjectionMatrix*gl_Vertex).xy;
    }

15  #endvertex

    uniform float Gamma;
    uniform float Exposure;
    uniform float Brightness;
20  uniform float Contrast;
    uniform float Saturation;
    uniform int ToneMapping;
    /*
    ** Based on: http://mouaif.wordpress.com/2009/01/22/photoshop-gamma-correction-%E2%99%A2shader/
    **
25  ** Contrast, saturation, brightness
    ** Code of this function is from TGM's shader pack
    ** http://irrlicht.sourceforge.net/phpBB2/viewtopic.php?t=21057
    */
30  // For all settings: 1.0 = 100% 0.5=50% 1.5 = 150%
    vec3 ContrastSaturationBrightness(vec3 color, float brt, float sat, float con)
    {
        const vec3 LumCoeff = vec3(0.2125, 0.7154, 0.0721);
        vec3 AvgLumin = vec3(0.5);
35        vec3 brtColor = color * brt;
        float intensityf = dot(brtColor, LumCoeff);
        vec3 intensity = vec3(intensityf, intensityf, intensityf);
        vec3 satColor = mix(intensity, brtColor, sat);
        vec3 conColor = mix(AvgLumin, satColor, con);
40        return conColor;
    }

    float sigmoid(float t) {
        float K = 1.0-1./(1.0+exp(-0.5*Contrast*5.));
45        t -= 0.5;
        float x = 1./(1.0+exp(-t*Contrast*5.))-K;
        return x/((1.0-2.0*K));
    }

50  vec3 sigmoid3(vec3 t) {
        return vec3(sigmoid(t.x),sigmoid(t.y),sigmoid(t.z));
    }

```

```

55   varying vec2 coord;
      uniform sampler2D frontbuffer;
      void main() {
          vec2 pos = (coord+vec2(1.0))/2.0;
60         vec4 tex = texture2D(frontbuffer, pos);
          vec3 c = tex.xyz/tex.a;

65         if (ToneMapping==1) {
            // Linear
            c = c*Exposure;
            c = ContrastSaturationBrightness(c, Brightness, Saturation, ↵
              ↵ Contrast);

70         } else if (ToneMapping==2) {
            // ExponentialExposure
            c = vec3(1.0)-exp(-c*Exposure);
            c = ContrastSaturationBrightness(c, Brightness, Saturation, ↵
              ↵ Contrast);

75         } else if (ToneMapping==3) {
            // Filmic: http://filmicgames.com/archives/75
            c*=Exposure;
            vec3 x = max(vec3(0.),c-vec3(0.004));
            c = (x*(6.2*x+.5))/(x*(6.2*x+1.7)+0.06);
80         c = pow(c, vec3(2.2)); // It already takes the Gamma into account
            c = ContrastSaturationBrightness(c, Brightness, Saturation, ↵
              ↵ Contrast);

            } else if (ToneMapping==4) {
                // Reinhart
85                 c*=Exposure;
                c = c/(1.+c);
                c = ContrastSaturationBrightness(c, Brightness, Saturation, ↵
                  ↵ Contrast);
            } else if (ToneMapping==5) {
                c = sigmoid3(c*Exposure+vec3(Brightness-1.0));
90            }

            c = pow(c, vec3(1.0/Gamma));

            gl_FragColor = vec4(c,1.0);
95     }

```

61 sketches/dolly/DE-Raytracer.frag

```

#donotrun
#include "3D.frag"

#group Raytracer
5 // Distance to object at which raymarching stops.
  uniform float Detail;slider[-7,-2.3,0];
  // The step size when sampling AO (set to 0 for old AO)
  uniform float DetailAO;slider[-7,-0.5,0];

```



```

10  const float ClarityPower = 1.0;

    // Lower this if the system is missing details
    uniform float FudgeFactor; slider[0,1,1];
15  float minDist = pow(10.0,Detail);
    float aoEps = pow(10.0,DetailAO);
    float MaxDistance = 100.0;

20  // Maximum number of raymarching steps.
    uniform int MaxRaySteps; slider[0,56,2000]

    // Use this to boost Ambient Occlusion and Glow
    //uniform float MaxRayStepsDiv; slider[0,1.8,10]
25  // Can be used to remove banding
    uniform float Dither; slider[0,0.5,1];

    // Used to prevent normals from being evaluated inside objects.
30  uniform float NormalBackStep; slider[0,1,10] Locked

#group Light

    // AO based on the number of raymarching steps
35  uniform vec4 AO; color[0,0.7,1,0.0,0.0,0.0];

    // The specular intensity of the directional light
    uniform float Specular; slider[0,0.4,1.0];
    // The specular exponent
40  uniform float SpecularExp; slider[0,16.0,100.0];
    // Limits the maximum specular strength to avoid artifacts
    uniform float SpecularMax; slider[0,10,100]
    // Color and strength of the directional light
    uniform vec4 SpotLight; color[0.0,0.4,1.0,1.0,1.0,1.0];
45  // Direction to the spot light (spherical coordinates)
    uniform vec2 SpotLightDir; slider[(-1,-1),(0.1,0.1),(1,1)]
    // Light coming from the camera position (diffuse lightning)
    uniform vec4 CamLight; color[0,1,2,1.0,1.0,1.0];
    // Controls the minimum ambient light, regardless of directionality
50  uniform float CamLightMin; slider[0.0,0.0,1.0]
    // Glow based on distance from fractal
    uniform vec4 Glow; color[0,0.0,1,1.0,1.0,1.0];
    //uniform vec4 InnerGlow; color[0,0.0,1,1.0,1.0,1.0];
    uniform int GlowMax; slider[0,20,1000]
55  // Adds fog based on distance
    uniform float Fog; slider[0,0.0,2]
    // Hard shadows shape is controlled by SpotLightDir
    uniform float HardShadow; slider[0,0,1] Locked

60  uniform float ShadowSoft; slider[0.0,2.0,20]

    uniform float Reflection; slider[0,0,1] Locked
    uniform bool DebugSun; checkbox[false] Locked
    vec4 orbitTrap = vec4(10000.0);
65  float fractionalCount = 0.0;

```

```

#group Coloring

// This is the pure color of object (in white light)
70 uniform vec3 BaseColor; color[1.0,1.0,1.0];
// Determines the mix between pure light coloring and pure orbit trap coloring
uniform float OrbitStrength; slider[0,0,1]

// Closest distance to YZ-plane during orbit
75 uniform vec4 X; color[-1,0.7,1,0.5,0.6,0.6];

// Closest distance to XZ-plane during orbit
uniform vec4 Y; color[-1,0.4,1,1.0,0.6,0.0];

80 // Closest distance to XY-plane during orbit
uniform vec4 Z; color[-1,0.5,1,0.8,0.78,1.0];

// Closest distance to origin during orbit
uniform vec4 R; color[-1,0.12,1,0.4,0.7,1.0];
85

// Background color
uniform vec3 BackgroundColor; color[0.6,0.6,0.45]
// Vignette background
uniform float GradientBackground; slider[0.0,0.3,5.0]
90

float DE(vec3 pos) ; // Must be implemented in other file

uniform bool CycleColors; checkbox[false]
uniform float Cycles; slider[0.1,1.1,32.3]
95

#ifdef providesNormal
vec3 normal(vec3 pos, float normalDistance);

#else
100 vec3 normal(vec3 pos, float normalDistance) {
    normalDistance = max(normalDistance*0.5, 1.0e-7);
    vec3 e = vec3(0.0,normalDistance,0.0);
    vec3 n = vec3(DE(pos+e.yxx)-DE(pos-e.yxx),
                  DE(pos+e.yyx)-DE(pos-e.yyx),
105                  DE(pos+e.xxy)-DE(pos-e.xxy));
    n = normalize(n);
    return n;
}
#endif
110

#ifdef providesBackground
vec3 backgroundColor(vec3 dir);
#endif

115 #group Floor

uniform bool EnableFloor; checkbox[false] Locked
uniform vec3 FloorNormal; slider[(-1,-1,-1),(0,0,1),(1,1,1)]
uniform float FloorHeight; slider[-5,0,5]
120 uniform vec3 FloorColor; color[1,1,1]
bool floorHit = false;
float floorDist = 0.0;
vec3 floorNormal = normalize(FloorNormal);

```

```

float fSteps = 0.0;
125 float DEF(vec3 p) {
    float d = DE(p);
    if (EnableFloor) {
        floorDist = abs(dot(floorNormal,p)-FloorHeight);
        if (d<floorDist) {
130             fSteps++;
            return d;
        } else return floorDist;
    } else {
        fSteps++;
135         return d;
    }
}

float DEF2(vec3 p) {
140     if (EnableFloor) {
        floorDist = abs(dot(floorNormal,p)-FloorHeight);
        return min(floorDist , DE(p));
    } else {
        return DE(p);
145     }
}

// Uses the soft-shadow approach by Quilez:
150 // http://iquilezles.org/www/articles/rmshadows/rmshadows.htm
float shadow(vec3 pos, vec3 sdir, float eps) {
    float totalDist =2.0*eps;
    float s = 1.0; // where 1.0 means no shadow!
    for (int steps=0; steps<MaxRaySteps && totalDist<MaxDistance; steps++) {
155         vec3 p = pos + totalDist * sdir;
        float dist = DEF2(p);
        if (dist < eps) return 1.0;
        s = min(s, ShadowSoft*pow((dist/totalDist),0.5));
        totalDist += dist;
160     }
    return 1.0-s;
}

float rand(vec2 co){
165     // implementation found at: lumina.sourceforge.net/Tutorials/Noise.html
    return fract(sin(dot(co.xy ,vec2(12.9898,78.233))) * 43758.5453);
}

vec3 lighting(vec3 n, vec3 color, vec3 pos, vec3 dir, float eps, out float &
    ↪ shadowStrength) {
170     shadowStrength = 0.0;
    vec3 spotDir = vec3(sin(SpotLightDir.x*3.1415)*cos(SpotLightDir.y&
        ↪ *3.1415/2.0), sin(SpotLightDir.y*3.1415/2.0)*sin(SpotLightDir.x&
        ↪ *3.1415), cos(SpotLightDir.x*3.1415));
    spotDir = normalize(spotDir);

    float nDotL = max(0.0,dot(n,spotDir));
175     vec3 halfVector = normalize(-dir+spotDir);
    float diffuse = nDotL*SpotLight.w;
    float ambient = max(CamLightMin,dot(-n, dir))*CamLight.w;

```

```

float hDotN = max(0., dot(n, halfVector));

180 // An attempt at Physcical Based Specular Shading:
// http://renderwonk.com/publications/s2010-shading-course/
// (Blinn-Phong with Schickl term and physical normalization)
float specular = ((SpecularExp+2.)/8.)*pow(hDotN, SpecularExp)*
185 (SpecularExp + (1.-SpecularExp)*pow(1.-hDotN, 5.))*
nDotL*Specular;
specular = min(SpecularMax, specular);

if (HardShadow>0.0) {
190 // check path from pos to spotDir
shadowStrength = shadow(pos+n*eps, spotDir, eps);
ambient = mix(ambient, 0.0, HardShadow*shadowStrength);
diffuse = mix(diffuse, 0.0, HardShadow*shadowStrength);
// specular = mix(specular, 0.0, HardShadow*f);
if (shadowStrength>0.0) specular = 0.0; // always turn off ↯
    ↯ specular, if blocked
195 }

return (SpotLight.xyz*diffuse+CamLight.xyz*ambient+ specular*SpotLight.↯
    ↯ xyz)*color;
}

200 vec3 colorBase = vec3(0.0,0.0,0.0);

vec3 cycle(vec3 c, float s) {
    return vec3(0.5)+0.5*vec3(cos(s*Cycles+c.x), cos(s*Cycles+c.y), cos(s*↯
        ↯ Cycles+c.z));
}

205 // Ambient occlusion approximation.
// Sample proximity at a few points in the direction of the normal.
float ambientOcclusion(vec3 p, vec3 n) {
    float ao = 0.0;
210 float de = DEF(p);
float wSum = 0.0;
float w = 1.0;
float d = 1.0-(Dither*rand(p.xy));
for (float i =1.0; i <6.0; i++) {
215 // D is the distance estimate difference.
// If we move 'n' units in the normal direction,
// we would expect the DE difference to be 'n' larger -
// unless there is some obstructing geometry in place
float D = (DEF(p+ d*n*i*i*aoEps) -de)/(d*i*i*aoEps);
220 w *= 0.6;
ao += w*clamp(1.0-D,0.0,1.0);
wSum += w;
}
return clamp(AO.w*ao/wSum, 0.0, 1.0);
225 }

vec3 getColor() {
    orbitTrap.w = sqrt(orbitTrap.w);

230 vec3 orbitColor;
if (CycleColors) {

```

```

        orbitColor = cycle(X.xyz,orbitTrap.x)*X.w*orbitTrap.x +
        cycle(Y.xyz,orbitTrap.y)*Y.w*orbitTrap.y +
        cycle(Z.xyz,orbitTrap.z)*Z.w*orbitTrap.z +
235     cycle(R.xyz,orbitTrap.w)*R.w*orbitTrap.w;
    } else {
        orbitColor = X.xyz*X.w*orbitTrap.x +
        Y.xyz*Y.w*orbitTrap.y +
        Z.xyz*Z.w*orbitTrap.z +
240     R.xyz*R.w*orbitTrap.w;
    }

    vec3 color = mix(BaseColor, 3.0*orbitColor, OrbitStrength);
    return color;
245 }

#ifdef providesColor
vec3 baseColor(vec3 point, vec3 normal);
#endif

250 vec3 trace(vec3 from, vec3 dir, inout vec3 hit, inout vec3 hitNormal) {
    hit = vec3(0.0);
    orbitTrap = vec4(10000.0);
    vec3 direction = normalize(dir);
255     floorHit = false;
    floorDist = 0.0;

    float dist = 0.0;
    float totalDist = 0.0;

260     int steps;
    colorBase = vec3(0.0,0.0,0.0);

265     // We will adjust the minimum distance based on the current zoom
    float eps = minDist;
    float epsModified = 0.0;

    for (steps=0; steps<MaxRaySteps; steps++) {
270         orbitTrap = vec4(10000.0);
        vec3 p = from + totalDist * direction;
        dist = DEF(p);
        dist *= FudgeFactor;

275         if (steps == 0) dist*=(Dither*rand(direction.xy))+(1.0-Dither);
        totalDist += dist;
        epsModified = pow(totalDist, ClarityPower)*eps;
        if (dist < epsModified) {
            // move back
280             totalDist -= (epsModified-dist);
            break;
        }
        if (totalDist > MaxDistance) {
            fSteps -= (totalDist-MaxDistance)/dist;
285             break;
        }
    }
    if (EnableFloor && dist == floorDist*FudgeFactor) floorHit = true;

```

```

vec3 hitColor;
290 float stepFactor = clamp((fSteps)/float(GlowMax),0.0,1.0);
vec3 backColor = BackgroundColor;
if (GradientBackground>0.0) {
    float t = length(coord);
    backColor = mix(backColor, vec3(0.0,0.0,0.0), t*↵
    ↵ GradientBackground);
295 }

if ( steps==MaxRaySteps) orbitTrap = vec4(0.0);

float shadowStrength = 0.0;
300 if ( dist < epsModified) {
    // We hit something, or reached MaxRaySteps
    hit = from + totalDist * direction;
    float ao = AO.w*stepFactor ;

305     if (floorHit) {
        hitNormal = floorNormal;
        if (dot(hitNormal,direction)>0.0) hitNormal *=-1.0;
    } else {
        hitNormal= normal(hit-NormalBackStep*epsModified*↵
        ↵ direction, epsModified); // /*normalE*epsModified/↵
        ↵ eps*/
310     }

    #ifdef providesColor
        hitColor = mix(BaseColor, baseColor(hit, hitNormal), ↵
        ↵ OrbitStrength);
315    #else
        hitColor = getColor();
    #endif
    #ifndef linearGamma
        hitColor = pow(clamp(hitColor,0.0,1.0),vec3(Gamma));
320    #endif
    if (DetailAO<0.0) ao = ambientOcclusion(hit, hitNormal);
    if (floorHit) {
        hitColor = FloorColor;
    }

325    hitColor = mix(hitColor, AO.xyz,ao);
    hitColor = lighting(hitNormal, hitColor, hit, direction,↵
    ↵ epsModified,shadowStrength);
    // OpenGL GL_EXP2 like fog
    float f = totalDist;
330    hitColor = mix(hitColor, backColor, 1.0-exp(-pow(Fog,4.0)*f*f));
    if (floorHit) {
        hitColor +=Glow.xyz*stepFactor* Glow.w*(1.0-↵
        ↵ shadowStrength);
    }
}
335 else {
    #ifdef providesBackground
        hitColor = backgroundColor(dir);
    #else
        hitColor = backColor;
    #endif
}

```

```

340         #endif
        hitColor +=Glow.xyz*stepFactor* Glow.w*(1.0-shadowStrength);
        hitNormal = vec3(0.0);
        if (DebugSun) {
            vec3 spotDir = vec3(sin(SpotLightDir.x*3.1415)*cos(↵
                ↵ SpotLightDir.y*3.1415/2.0), sin(SpotLightDir.y↵
                ↵ *3.1415/2.0)*sin(SpotLightDir.x*3.1415), cos(↵
                ↵ SpotLightDir.x*3.1415));
345        spotDir = normalize(spotDir);
        if (dot(spotDir, normalize(dir))>0.9) hitColor= vec3↵
            ↵ (100.,0.,0.);
        }
    }

350    return hitColor;
}

vec3 color(vec3 from, vec3 dir) {
    vec3 hit = vec3(0.0);
355    vec3 hitNormal = vec3(0.0);
    if (Reflection==0.) {
        return trace(from, dir, hit, hitNormal);
    } else {
        vec3 first = trace(from, dir, hit, hitNormal);
360        if (hitNormal == vec3(0.0)) return first;
        vec3 d = reflect(dir, hitNormal);
        return mix(first, trace(hit+d*minDist, d, hit, hitNormal),↵
            ↵ Reflection);
    }
}

```

62 sketches/dolly/dolly.frag

```

#define providesColor

#include "DE-Raytracer.frag"

5    vec3 baseColor(vec3 p, vec3 n) {
    if (length(p - Target) < 0.11) { return vec3(1.0, 0.7, 0.0); } else { return ↵
        ↵ vec3(0.0, 1.0, 0.7); }
    }

    float DE(vec3 p) {
10    return min(p.y + 0.01 * (cos(3.0 * p.x) * cos(4.0 * p.z) + cos(5.0 * (p.z - p.↵
        ↵ x)) * cos(2.0 * (p.z + p.x))) * max(0.1, p.z * p.z / 20.0), length(p - ↵
        ↵ Target) - 0.1);
    }

```

63 sketches/juggling/jugglee-help.pd

```

#N canvas 0 0 450 414 10;
#X obj 242 126 jugglee;
#X obj 242 79 juggler;
#X obj 324 137 print juggler;
5 #X obj 242 41 t b b;
#X obj 300 22 loadbang;

```

```

#X obj 134 26 mores 1;
#X obj 189 52 - 1;
#X obj 187 27 t f b;
10 #X msg 296 183 create \, 1;
#X msg 311 210 0 \, destroy;
#X obj 16 23 gemhead;
#X obj 17 95 t b a;
#X obj 13 134 gemlist;
15 #X obj 242 150 route ball;
#X obj 242 171 t b a;
#X obj 111 197 unpack f f f;
#X obj 109 282 pack f 1 1;
#X obj 109 305 Gem/hsv2rgb;
20 #X obj 111 239 mod 360;
#X obj 111 260 / 360;
#X obj 180 246 pack f f 0;
#X obj 15 239 translate;
#X obj 15 336 translate;
25 #X obj 49 163 loadbang;
#X msg 48 307 -1;
#X msg 41 187 1;
#X obj 15 260 color;
#X obj 15 281 sphere 0.1;
30 #X obj 111 218 * 222.492;
#X obj 201 223 - 3;
#X obj 324 112 spigot;
#X obj 356 86 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
1;
35 #X obj 172 224 * 2;
#X obj 199 203 mores 0;
#X msg 300 47 balls 5 \, throw 7;
#X obj 234 225 / 2;
#X obj 73 33 + 0.1;
40 #X obj 296 236 gemwin 25;
#X obj 70 12 f 0.05;
#X connect 0 0 13 0;
#X connect 1 0 0 0;
#X connect 1 0 30 0;
45 #X connect 3 0 1 0;
#X connect 3 1 0 0;
#X connect 4 0 34 0;
#X connect 5 0 0 0;
#X connect 5 0 38 1;
50 #X connect 5 1 7 0;
#X connect 6 0 5 0;
#X connect 7 0 6 0;
#X connect 7 1 3 0;
#X connect 8 0 37 0;
55 #X connect 9 0 37 0;
#X connect 10 0 11 0;
#X connect 11 0 38 0;
#X connect 11 1 12 1;
#X connect 12 0 21 0;
60 #X connect 13 0 14 0;
#X connect 14 0 12 0;
#X connect 14 1 15 0;
#X connect 15 0 28 0;

```



```

#X connect 15 1 32 0;
65 #X connect 15 2 33 0;
#X connect 16 0 17 0;
#X connect 17 0 26 1;
#X connect 18 0 19 0;
#X connect 19 0 16 0;
70 #X connect 20 0 21 2;
#X connect 20 0 22 2;
#X connect 21 0 26 0;
#X connect 23 0 24 0;
#X connect 23 0 25 0;
75 #X connect 24 0 22 1;
#X connect 25 0 21 1;
#X connect 26 0 27 0;
#X connect 27 0 22 0;
#X connect 28 0 18 0;
80 #X connect 29 0 20 1;
#X connect 30 0 2 0;
#X connect 31 0 30 1;
#X connect 32 0 20 0;
#X connect 33 0 29 0;
85 #X connect 33 1 35 0;
#X connect 34 0 1 0;
#X connect 35 0 29 0;
#X connect 36 0 5 0;
#X connect 38 0 36 0;

```

64 sketches/juggling/jugglee.pd_lua

```

local J = pd.Class:new():register("jugglee")

function J:initialize()
    self.balls = { }
    5 self.currentBeat = 0
    self.currentHand = 1
    self.throwTime = 0.2
    self.catchTime = 0.2
    self.handRadius = 0.2
    10 self.inlets = 1
    self.outlets = 1
    return true
end

15 local function position(hands, ball, t)
    if (ball.t1 - ball.t0) == 1 then
        if t < ball.t0 then
            return { 0, -1 }
        elseif t < ball.t1 then
            20 return { ball.x0 + (ball.x1 - ball.x0) * (t - ball.t0), -hands.handRadius }
        else
            return { 0, -1 }
        end
    else
        25 if t < ball.t0 then
            return { 0, -1 }
        elseif t < ball.t0 + hands.throwTime then

```

```

        local d = 1
        if ball.x0 < 0 then d = -1 end
30      local r = hands.handRadius
        local a = math.pi/2 * (t - ball.t0) / hands.throwTime
        return { ball.x0 + d * r * (1 - math.sin(a)), -r * math.cos(a) }
    elseif t < ball.t1 - hands.catchTime then
        local throw = (ball.t1 - hands.catchTime) - (ball.t0 + hands.throwTime)
35      local dt = (t - (ball.t0 + hands.throwTime))
        return { ball.x0 + (ball.x1 - ball.x0) * dt / throw, throw * dt - dt * dt ↵
            ↵ }
    elseif t < ball.t1 then
        local d = 1
        if ball.x1 < 0 then d = -1 end
40      local r = hands.handRadius
        local a = math.pi/2 * (ball.t1 - t) / hands.catchTime
        return { ball.x1 - d * r * (1 - math.sin(a)), -r * math.cos(a) }
    else
        return { 0, -1 }
45    end
  end
end

function J:in_1_bang()
50  self.currentBeat = self.currentBeat + 1
  self.currentHand = -self.currentHand
  local t = self.currentBeat + 0.5
  local balls = { }
  local i
55  for i = 1,#(self.balls) do
    local ball = self.balls[i]
    local p = position(self, ball, t)
    if p[2] > 0 then
      table.insert(balls, ball)
60    end
  end
  self.balls = balls
end

65  function J:in_1_float(phase)
    local t = self.currentBeat + phase
    local i
    for i = 1,#(self.balls) do
      local ball = self.balls[i]
70      local p = position(self, ball, t)
      local x = p[1]
      local y = p[2]
      self:outlet(1, "ball", { ball.id, x, y })
    end
75  end

function J:in_1_throw(atoms)
  local id = atoms[1]
  local direction = atoms[2]
80  local speed = atoms[3]
  local fhand = self.currentHand
  local thand = self.currentHand + 2 * direction
  local dx = -direction

```

```

    local ball = { }
85   ball.id = id
      ball.x0 = fhand * (1 - self.handRadius)
      ball.x1 = thand * (1 + self.handRadius)
      ball.t0 = self.currentBeat
      ball.t1 = self.currentBeat + speed
90   table.insert(self.balls, ball)
end

function J:in_1_pick() end
function J:in_1_drop() end
95  function J:in_1_wait() end
      function J:in_1_pause() end
      function J:in_1_remove() end

```

65 sketches/juggling/juggler-help.pd

```

#N canvas 0 0 450 300 10;
#X obj 105 131 juggler;
#X msg 139 75 balls 3 \, throw 5;
#X obj 107 152 print juggle;
5  #X obj 104 75 bng 15 250 50 0 empty empty empty 17 7 0 10 -262144 -1
    -1;
#X msg 166 96 balls 4;
#X msg 184 120 balls 8;
#X msg 260 102 throw 3;
10 #X msg 260 134 throw 10;
#X msg 204 163 balls 0;
#X connect 0 0 2 0;
#X connect 1 0 0 0;
#X connect 3 0 0 0;
15 #X connect 4 0 0 0;
#X connect 5 0 0 0;
#X connect 6 0 0 0;
#X connect 7 0 0 0;
#X connect 8 0 0 0;

```

66 sketches/juggling/juggler.pd_lua

```

local J = pd.Class:new():register("juggler")

function J:initialize()
    self.currentHand = -1
5   self.nextBallID = 0
      self.currentBallCount = 0
      self.targetBallCount = 0
      self.maximumThrow = 0
      self.longestThrow = 0
10  self.futureLandings = { }
      self.inlets = 1
      self.outlets = 1
      return true
end
15

function J:in_1_balls(atoms)
    if type(atoms[1] == "number") then

```

```

        self.targetBallCount = math.max(0, math.floor(atoms[1]))
    end
20 end

function J:in_1_throw(atoms)
    if type(atoms[1] == "number") then
        self.maximumThrow = math.max(0, math.floor(atoms[1]))
25     self.longestThrow = math.max(self.longestThrow, self.maximumThrow)
    end
end

function J:in_1_bang()
30     -- catch a possible ball
    local caught = self.futureLandings[1]
    -- advance futures
    local n = self.longestThrow
    local i
35     for i = 1,n do
        self.futureLandings[i] = self.futureLandings[i + 1]
    end
    -- check what we caught
    if type(caught) == "number" then
40         -- a ball
        if self.currentBallCount > self.targetBallCount then
            -- remove it
            self.currentBallCount = self.currentBallCount - 1
            self:outlet(1, "remove", { caught })
45         else
            -- collect possible throws without landing collisions
            local choices = { }
            for i = 1,self.maximumThrow do
                if type(self.futureLandings[i]) ~= "number" then
50                     table.insert(choices, i)
                end
            end
            -- choose a legal throw
            if #choices == 0 then
55                 self.currentBallCount = self.currentBallCount - 1
                self:outlet(1, "drop", { caught })
            else
                local dice = math.random()
--[[--
60         dice = dice * dice
        if self.currentHand < 0 then
            dice = 1 - dice
        end
--]]--
65         dice = math.ceil(dice * #choices)
        dice = math.min(math.max(dice, 1), #choices)
        local throw = choices[dice]
        -- check direction of throw
        local direction
70         if (throw / 2) == math.floor(throw / 2) then
            direction = 0
        elseif self.currentHand < 0 then
            direction = 1
        else

```

```

75         direction = -1
        end
        -- schedule future landing
        self.futureLandings[throw] = caught
        self:outlet(1, "throw", { caught, direction, throw })
80     end
    end
  else
    -- caught nothing
    if self.currentBallCount < self.targetBallCount then
85      -- collect possible throws without landing collisions
      local choices = { }
      for i = 1, self.maximumThrow do
        if type(self.futureLandings[i]) ~= "number" then
          table.insert(choices, i)
90      end
        end
        -- choose a legal throw
        if #choices == 0 then
          self:outlet(1, "wait", { })
95      else
        local choice = math.random(1, #choices)
        local throw = choices[choice]
        -- check direction of throw
        local direction
100      if (throw / 2) == math.floor(throw / 2) then
        direction = 0
        elseif self.currentHand < 0 then
          direction = 1
        else
105      direction = -1
        end
        -- introduce a new ball
        local ball = self.nextBallID
        self.nextBallID = self.nextBallID + 1
110      self.currentBallCount = self.currentBallCount + 1
        -- schedule future landing
        self.futureLandings[throw] = ball
        self:outlet(1, "pick", { ball })
        self:outlet(1, "throw", { ball, direction, throw })
115      end
    else
      -- no action needed
      self:outlet(1, "pause", { })
    end
  end
120 end
  -- swap hands
  self.currentHand = -self.currentHand
end

```

67 sketches/lightning/.gitignore

```

lightning
*.ppm

```

68 sketches/lightning/lightning.c

```

#include <limits.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
5  #include <time.h>

#define width 256
#define height 256

10  int sky[height][width];

    unsigned char ppm[height][width][3];

    static inline int min(int i, int j) { return i < j ? i : j; }
15  static inline int max(int i, int j) { return i > j ? i : j; }
    static inline int clamp(int i, int lo, int hi) { return min(max(i, lo), hi); }

    int strike(int j, int i) {
        int n = sky[j][i];
20        if (n > 0) {
            for (int jj = j - 1; jj <= j + 1; ++jj) {
                for (int ii = i - 1 + width; ii <= i + 1 + width; ++ii) {
                    int iii = ii % width;
                    int jjj = clamp(jj, 0, height - 1);
25                    if (sky[jjj][iii] == n - 1) {
                        if (strike(jjj, iii)) {
                            ppm[jjj][iii][0] = 255;
                            ppm[jjj][iii][1] = 255;
                            return 1;
30                }
            }
        }
        }
        /*else if (n < -1) {
35        for (int jj = j - 1; jj <= j + 1; ++jj) {
            for (int ii = i - 1 + width; ii <= i + 1 + width; ++ii) {
                int iii = ii % width;
                if (sky[jj][iii] == n + 1) {
                    if (strike(jj, iii)) {
40                        ppm[jj][iii][0] = 255;
                        ppm[jj][iii][1] = 255;
                        return 1;
                    }
                }
            }
45        }
        }
        */else {
            return 1;
        }
50    return 0;
}

int main(int argc, char **argv) {
    srand(time(0));
55    for (int j = 0; j < height; ++j) {
        for (int i = 0; i < width; ++i) {
            sky[j][i] = j == 0;

```

```

    }
}
60 int n, h = 0, i, j, k = 0, complete = 0;
do {
    do {
        j = h + 1;
        i = rand() % width;
65     do {
        j += rand() % 5 - 2;
        i += rand() % 5 - 2 + width;
        i %= width;
        k++;
70     n = 0;
        j = clamp(j, 0, min(h + 4, height - 1));
        for (int jj = j - 1; jj <= j + 1; ++jj) {
            for (int ii = i - 1; ii <= i + 1; ++ii) {
                if (sky[clamp(jj, 0, height-1)][(ii + width) % width]) {
75                 if (n) {
                    n = min(n, sky[clamp(jj, 0, height-1)][(ii + width) % width]);
                } else {
                    n = sky[clamp(jj, 0, height-1)][(ii + width) % width];
                }
            }
80         }
        }
    } while (! n);
} while (! n);
85 /*
    int mi = INT_MAX;
    int ma = INT_MIN;
    for (int jj = j - 1; jj <= j + 1; ++jj) {
        for (int ii = i - 1; ii <= i + 1; ++ii) {
90         //mi = min(mi, sky[clamp(jj, 0, height-1)][(ii + width) % width]);
        ma = max(ma, sky[clamp(jj, 0, height-1)][(ii + width) % width]);
        }
    }
*/
95 sky[j][i] = n + 1;
/*
    if (mi < 0 && 0 < ma) {
    } else if (mi < 0) {
        sky[j][i] = mi - 1;
100    } else if (0 < ma) {
        sky[j][i] = ma + 1;
    }
*/
    h = max(h, j);
105    complete = h == height - 1;
} while (! complete);
memset(&ppm[0][0][0], 0, width * height * 3);
strike(j, i);
for (int j = 0; j < height; ++j) {
110    for (int i = 0; i < width; ++i) {
        ppm[j][i][2] = sky[j][i] ? 255 : 0;
    }
}
printf("P6\n%d %d\n255\n", width, height);

```

```

115    fwrite(&ppm[0][0][0], width * height * 3, 1, stdout);
        fprintf(stderr, "%d\n", k);
        return 0;
    }

```

69 sketches/lightning/lightning.sh

```

#!/bin/bash
make
for i in $(seq 0 9)
do
5    ./lightning
    sleep 2
done |
pnmsplit - lightning-%d.ppm

```

70 sketches/lightning/Makefile

```

lightning: lightning.c
        gcc -std=c99 -Wall -pedantic -Wextra -O3 -o lightning lightning.c

```

71 sketches/mountains/.gitignore

```

mountains
*.pgm

```

72 sketches/mountains/Makefile

```

mountains: mountains.c
        gcc -std=c99 -Wall -pedantic -Wextra -O3 -o mountains mountains.c -x
        ↪ fopenmp

```

73 sketches/mountains/mountains.c

```

#include <limits.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
5  #include <time.h>

#define width 1024
#define height 1024

10  float mountain[width];

unsigned char pgm[height][width];

static inline int min(int i, int j) { return i < j ? i : j; }
15  static inline int max(int i, int j) { return i > j ? i : j; }
static inline int clamp(int i, int lo, int hi) { return min(max(i, lo), hi); }

void subdivide(int i0, int i1, float h0, float h1, float dh) {
    if (i0 >= i1 - 1) {
20        mountain[i0] = h0;

```



```

    } else {
        int i2 = (i0 + i1) / 2;
        float h2 = (h0 + h1) / 2 + dh * (rand() / (double) RAND_MAX - 0.5) * 2;
        subdivide(i0, i2, h0, h2, dh * 0.7);
25     subdivide(i2, i1, h2, h1, dh * 0.7);
    }
}

int main(int argc, char **argv) {
30     srand(time(0));
    for (int k = 0; k < 10; ++k) {
        subdivide(0, width, height/2, height/2, height/4);
        memset(&pgm[0][0], 0, width * height);
        #pragma omp parallel for
35     for (int j = 0; j < height; ++j) {
        for (int i = 0; i < width; ++i) {
            int n = 0;
            for (int jj = j - 1; jj <= j + 1; ++jj) {
                for (int ii = i - 1; ii <= i + 1; ++ii) {
40                     n += clamp(jj, 0, height - 1) < mountain[(ii + width) % width];
                }
            }
            pgm[j][i] = (0 < n && n < 9) * 255 + (n < 9) * 0;
        }
45     }
    printf("P5\n%d %d\n255\n", width, height);
    fwrite(&pgm[0][0], width * height, 1, stdout);
}
return 0;
50 }

```

74 sketches/mountains/mountains.sh

```

#!/bin/bash
make
./mountains |
pnmsplit - mountains-%d.pgm

```

75 sketches/oloid/.gitignore

```
oloid.obj
```

76 sketches/oloid/Makefile

```

oloid.obj: oloid.hs
    runghc oloid.hs

```

77 sketches/oloid/oloid.hs

```

import Data.Monoid (mconcat)

type R = Double

5  type V = (R, R, R)

```

```

norm2 (x, y, z) = x*x + y*y + z*z
norm x = sqrt (norm2 x)
midpoint = mix 0.5
10 mix k (x, y, z) (u, v, w) = (x * k + k' * u, y * k + k' * v, z * k + k' * w) ↯
    ↳ where k' = 1 - k
distance (x, y, z) (u, v, w) = norm (x - u, y - v, z - w)

l = sqrt 3

15 -- ends of torse parameterized by angle
a t = (sin t, -1/2 - cos t, 0)
b t = (0, 1/2 + cos t, sin t)

-- angles on the same torse of length l satisfy
20 -- cos^2(u/2) cos^2(t/2) = 1/2^2
other t = 2 * acos (1 / (2 * cos (t / 2)))

-- midpoint of a torse
m t = midpoint (a t) (b (other t))
25

-- surface area
area = 4 * pi
triangleCount = 2 * 4 * 10 * 16
triangleArea = area / triangleCount
30

-- circumference through midpoint of torse
circumference = 4 * sum (zipWith distance ms (tail ms))
    where ms = [ m t | i <- [0 .. 1200], let t = 2 * pi * fromIntegral i / 3600 ]

35 -- parameterize circumference
c s = sum (zipWith distance ms (tail ms)) + (s' - fromIntegral (floor s')) * ↯
    ↳ distance (m (fromIntegral (ceiling s') * 2 * pi / 3600)) (m (fromIntegral ↯
    ↳ (floor s') * 2 * pi / 3600))
    where
        ms = [ m t | i <- [0 .. floor s'], let t = 2 * pi * fromIntegral i / 3600 ]
        s' = 3600 * s / (2 * pi)
40

-- binary search
search eps f target lo mid hi
    | abs (f mid - target) < eps = mid
    | f mid < target = search eps f target mid ((mid + hi)/2) hi
45 | otherwise = search eps f target lo ((lo + mid)/2) mid

e = circumference / 180

ts = [ search e c t 0 (pi / 3) (2 * pi / 3) | i <- [30, 90 .. 1200], let t = 2 * ↯
    ↳ pi * (fromIntegral i) / 3600 ]
50 us = map other ts

mesh = triangleStrip $ concat
    [ zip ts us
    , zip (reverse ts) (map negate $ reverse us)
55 , zip (map negate ts) (map negate us)
    , zip (map negate $ reverse ts) (reverse us)
    ]

triangleStrip params@((t0, u0):_) = go params

```

```

60   where
      go [] = a t0 : b u0 : []
      go ((t, u):tus) = a t : b u : go tus

triangles (a0:b0:ab@(a1:b1:_)) = subdivide a0 a1 b0 b1 ++ triangles ab
65   where
      subdivide a0 a1 b0 b1
        | lab * l01 > 8 * triangleArea = subdivide a0 a1 c0 c1 ++ subdivide c0 c1 ↗
          ↘ b0 b1
        | la == 0 = [(b0,a1,b1)]
        | lb == 0 = [(a0,a1,b0)]
70   | otherwise = [(a0,a1,b0),(b0,b1,a1)]
      where
        lc = sqrt ((la * la + lb * lb) / 2)
        k = (la + lb) / (2 * (la + lc))
        c0 = mix k a0 b0
75   c1 = mix k a1 b1
        l0 = distance a0 b0
        l1 = distance a1 b1
        la = distance a0 a1
        lb = distance b0 b1
80   l01 = l0 + l1
        lab = la + lb
      triangles _ = []

obj s = let ~(vs, fs) = mconcat $ go 1 s in unlines (vs ++ fs)
85   where
      go - [] = []
      go i ((a, b, c):abc) = ([v a, v b, v c], [t i]) : go (i + 3) abc
      v (x, y, z) = "v " ++ show x ++ " " ++ show y ++ " " ++ show z
      t i = "f " ++ show i ++ " " ++ show (i + 1) ++ " " ++ show (i + 2)
90   (xs, ys, zs) = unzip3 mesh

main = do
  writeFile "oloid.obj" . obj . triangles $ mesh
95  {-
    print (length mesh)
    writeFile "oloid-xs.txt" (unlines (map show xs))
    writeFile "oloid-ys.txt" (unlines (map show ys))
    writeFile "oloid-zs.txt" (unlines (map show zs))
100   writeFile "oloid-rs.txt" (unlines (map show (map (\x -> 0.5 * x + 0.5) xs)))
    writeFile "oloid-gs.txt" (unlines (map show (map (\y -> y / 3 + 0.5) ys)))
    writeFile "oloid-bs.txt" (unlines (map show (map (\x -> 0.5 * x + 0.5) zs)))
    writeFile "oloid-as.txt" (unlines (map show (map (const 1) zs)))
  -}

```

78 sketches/oloid/oloid.pd

```

#N canvas 1339 393 652 448 10;
#X obj 11 10 gemhead;
#X obj 330 208 table oloid-xs 162;
#X obj 330 229 table oloid-ys 162;
5  #X obj 330 250 table oloid-zs 162;
#X obj 440 407 gemvertexbuffer 162;
#X msg 229 122 \; oloid-\$1s read oloid-\$1s.txt;
#X obj 229 97 symbol;

```

```

#X msg 121 31 0 \, destroy;
10 #X obj 65 127 f 0;
#X obj 114 128 + 1;
#X obj 454 306 t a b a b;
#X obj 421 354 colorRGB 0 0 0;
#X obj 47 92 t a b;
15 #X obj 49 216 rotateXYZ 0 0 0;
#X obj 421 333 scale 1.1;
#X obj 450 208 table oloid-rs 162;
#X obj 450 229 table oloid-gs 162;
#X obj 450 250 table oloid-bs 162;
20 #X msg 556 354 draw tristrip \, color_enable 1;
#X msg 229 76 x \, y \, z \, r \, g \, b \, a;
#X obj 450 270 table oloid-as 162;
#X msg 161 159 position oloid-xs oloid-ys oloid-zs \, color oloid-rs
oloid-gs oloid-bs oloid-as;
25 #X msg 486 330 draw line \, color_enable 0;
#X obj 16 406 pix_write;
#X obj 114 149 mod 720;
#X obj 117 175 / 2;
#X obj 65 178 * 2;
30 #X msg 33 358 auto \$1;
#X obj 31 327 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
1;
#X obj 111 104 gemwin 50;
#X obj 134 247 spigot 0;
35 #X obj 196 247 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
1;
#X obj 35 248 spigot 1;
#X obj 104 249 ==;
#X obj 92 375 GEMglPolygonMode GLFRONT_AND_BACK GL_LINE;
40 #X obj 18 158 scale 2.5;
#X obj 11 31 t a a a;
#X obj 34 53 separator;
#X obj 51 297 scale 1.01;
#X obj 35 269 t a a a;
45 #X obj 143 353 GEMglPolygonMode GLFRONT_AND_BACK GL_FILL;
#X msg 110 11 dimen 720 720 \, create \, 1 \, lighting 1;
#X obj 54 324 colorRGB 1 1 1;
#X obj 399 20 GEMglDisable GL_CULL_FACE;
#X obj 141 420 model;
50 #X obj 572 28 separator;
#X obj 570 76 world_light 1;
#X obj 570 126 world_light 2;
#X obj 571 175 world_light 3;
#X obj 571 225 world_light 4;
55 #X msg 723 54 1;
#X obj 569 54 rotateXYZ 0 45 45;
#X obj 569 104 rotateXYZ 0 45 45;
#X obj 569 153 rotateXYZ 0 45 45;
#X obj 570 203 rotateXYZ 0 45 45;
60 #X obj 38 72 GEMglShadeModel GLSMOOTH;
#X obj 164 272 shininess 50;
#X msg 720 92 1 1 0;
#X msg 719 127 1 0 1;
#X msg 722 170 1 1 0;
65 #X msg 700 205 1 1 1;

```

```
#X obj 173 330 specularRGB 0.5 0.5 1;
#X obj 173 309 ambientRGB 0.5 0.5 0.5;
#X obj 173 288 diffuseRGB 0.5 0.25 0.125;
#X msg 165 396 open oloid.obj \, smooth 0.5;
70 #X msg 86 94 0;
#X obj 94 50 bng 15 250 50 0 empty empty empty 17 7 0 10 -262144 -1
-1;
#X obj 201 419 loadbang;
#X obj 202 48 bng 15 250 50 0 empty empty empty 17 7 0 10 -262144 -1
75 -1;
#X connect 0 0 36 0;
#X connect 6 0 5 0;
#X connect 7 0 29 0;
#X connect 8 0 9 0;
80 #X connect 8 0 13 2;
#X connect 8 0 25 0;
#X connect 8 0 26 0;
#X connect 9 0 24 0;
#X connect 10 0 14 0;
85 #X connect 10 1 22 0;
#X connect 10 2 4 0;
#X connect 10 3 18 0;
#X connect 11 0 4 0;
#X connect 12 0 35 0;
90 #X connect 12 1 8 0;
#X connect 13 0 30 0;
#X connect 13 0 32 0;
#X connect 14 0 11 0;
#X connect 18 0 4 0;
95 #X connect 19 0 6 0;
#X connect 21 0 4 0;
#X connect 22 0 4 0;
#X connect 24 0 8 1;
#X connect 25 0 13 3;
100 #X connect 26 0 13 1;
#X connect 27 0 23 0;
#X connect 28 0 27 0;
#X connect 30 0 10 0;
#X connect 31 0 30 1;
105 #X connect 31 0 33 0;
#X connect 32 0 39 0;
#X connect 33 0 32 1;
#X connect 34 0 44 0;
#X connect 35 0 13 0;
110 #X connect 36 0 23 0;
#X connect 36 1 37 0;
#X connect 36 2 45 0;
#X connect 37 0 55 0;
#X connect 38 0 42 0;
115 #X connect 39 2 56 0;
#X connect 40 0 44 0;
#X connect 41 0 29 0;
#X connect 42 0 34 0;
#X connect 45 0 51 0;
120 #X connect 46 0 52 0;
#X connect 47 0 53 0;
#X connect 48 0 54 0;
```

```

#X connect 50 0 46 0;
#X connect 50 0 47 0;
125 #X connect 50 0 48 0;
#X connect 50 0 49 0;
#X connect 51 0 46 0;
#X connect 52 0 47 0;
#X connect 53 0 48 0;
130 #X connect 54 0 49 0;
#X connect 55 0 12 0;
#X connect 56 0 63 0;
#X connect 57 0 46 1;
#X connect 58 0 47 1;
135 #X connect 59 0 48 1;
#X connect 60 0 49 1;
#X connect 61 0 40 0;
#X connect 62 0 61 0;
#X connect 63 0 62 0;
140 #X connect 64 0 44 0;
#X connect 65 0 8 1;
#X connect 66 0 65 0;
#X connect 66 0 28 0;
#X connect 67 0 64 0;
145 #X connect 68 0 21 0;
#X connect 68 0 19 0;

```

79 sketches/prism/.gitignore

```

prism.position
prism.normal

```

80 sketches/prism/Makefile

```

prism.position prism.normal: prism.hs
    runghc prism.hs

```

81 sketches/prism/prism.frag

```

#version 120

uniform vec3 ambient;
uniform vec3 diffuse;
5 uniform vec3 specular;
uniform float shine;

uniform vec3 light;

10 varying vec4 position;
varying vec3 normal;

void main(void) {
    vec3 p = position.xyz/position.w;
15 vec3 n = normalize(normal);
    vec3 l = normalize(light - p);
    float dlm = max(dot(l, n), 0.0);
    vec3 reflected = normalize(reflect(-l, n));
    float highlight = pow(max(reflected.z, 0.0), shine);

```

```

20    gl_FragColor = vec4(mix(ambient + diffuse * dlm, specular, highlight), 1.0);
    }

```

82 sketches/prism/prism-help.pd

```

#N canvas 0 0 566 317 10;
#X obj 82 196 prism;
#X obj 82 34 gemhead;
#X obj 82 55 t a b;
5  #X obj 108 80 f 0;
#X obj 171 75 + 1;
#X obj 171 96 mod 360;
#X obj 108 110 * 8;
#X obj 407 154 gemwin 25;
10 #X msg 275 82 0 \, destroy;
#X obj 82 139 rotateXYZ 0 0 30;
#X msg 276 34 dimen 640 480 \, create \, 1 \, view 0 0 20 \, perspec
-1 1 -1 1 4 50;
#X obj 82 160 scale 3;
15 #X obj 201 134 / 90;
#X obj 196 179 expr min(max(if($f1 > 2 \, 3 - $f1 \, $f1) \, 0) \,
1);
#X obj 42 247 pix_write;
#X msg 64 222 auto 1;
20 #X obj 37 86 t a a;
#X obj 23 203 t b a;
#X msg 141 52 -1;
#X obj 15 171 spigot;
#X obj 50 141 tgl 15 0 empty empty empty 17 7 0 10 -262144 -1 -1 0
25 1;
#X connect 1 0 2 0;
#X connect 2 0 16 0;
#X connect 2 1 3 0;
#X connect 3 0 4 0;
30 #X connect 3 0 6 0;
#X connect 3 0 12 0;
#X connect 4 0 5 0;
#X connect 5 0 3 1;
#X connect 6 0 9 2;
35 #X connect 8 0 7 0;
#X connect 9 0 11 0;
#X connect 10 0 7 0;
#X connect 11 0 0 0;
#X connect 12 0 13 0;
40 #X connect 13 0 0 1;
#X connect 15 0 14 0;
#X connect 16 0 19 0;
#X connect 16 1 9 0;
#X connect 17 0 14 0;
45 #X connect 17 1 14 0;
#X connect 18 0 3 1;
#X connect 19 0 17 0;
#X connect 20 0 19 1;

```

83 sketches/prism/prism.hs

```

import Data.List (transpose)

type R = Double

5  type P = [R]

    midpoint ps qs = map (/2) $ zipWith (+) ps qs

    type N = [R]
10  magnitude = sqrt . sum . map (^2)

    normalize ps = map (/ magnitude ps) ps

15  midnormal ps qs = normalize $ zipWith (+) ps qs

    data V = V P N

    midvertex (V p m) (V q n) = V (midpoint p q) (midnormal m n)
20  sphere (V p _) = V q q where q = normalize p

    data T = T V V V

25  mapT f (T a b c) = T (f a) (f b) (f c)

    subdivide (T a b c) =
      [ T a ab ca
        , T ab b bc
30    , T ca bc c
        , T ab bc ca
      ]
      where
        ab = midvertex a b
        bc = midvertex b c
35    ca = midvertex c a

    subdivides = iterate (concatMap subdivide)

40  triangle a b c = T (V a n) (V b n) (V c n)
      where n = normalize $ (a 'to' b) 'cross' (a 'to' c)

    quad a b c d = [ triangle a b c, triangle c d a ]

45  to = zipWith subtract

    cross [u1,u2,u3] [v1,v2,v3] = [u2*v3 - u3*v2, u3*v1 - u1*v3, u1*v2 - u2*v1 ]

    prism l
50    = triangle a b c
      : triangle d f e
      : quad a c f d
      ++ quad c b e f
      ++ quad b a d e
55    where
      a = [-1 / 2, 1 / 2, 1 * sqrt 3 / 6]
      b = [ 1 / 2, 1 / 2, 1 * sqrt 3 / 6]

```



```

        c = [ 0 , 1 / 2, -1 * sqrt 3 / 2]
        d = [-1 / 2, -1 / 2, 1 * sqrt 3 / 6]
60      e = [ 1 / 2, -1 / 2, 1 * sqrt 3 / 6]
        f = [ 0 , -1 / 2, -1 * sqrt 3 / 2]

    toList (T (V a p) (V b q) (V c r)) = [ a ++ b ++ c, p ++ q ++ r]

65  tables = foldr (zipWith (++)) (repeat []) . map toList

    writeTable name xs = writeFile name (unlines $ map show xs)

main = do
70  let volume = ((4 * pi / 3) / (sqrt 3 / 4)) ** (1 / 3)
        -- area = ((4 * pi) / (3 + sqrt 3 / 2)) ** (1 / 2)
        -- circum = sqrt (12 / 7)
        [position, normal] = tables $ subdivides (prism volume) !! 3
    writeTable "prism.position" position
75  writeTable "prism.normal" normal

```

84 sketches/prism/prism.pd

```

#N canvas 0 0 521 360 10;
#X obj 40 24 inlet;
#X obj 40 45 glsl_vertex prism.vert;
#X obj 40 67 glsl_fragment prism.frag;
5  #X obj 40 213 glsl_program;
#X obj 298 67 pack f f;
#X obj 209 25 inlet;
#X msg 91 116 factor \ $1;
#X obj 185 277 table \ $0-position 4608;
10 #X obj 185 298 table \ $0-normal 4608;
#X obj 337 176 loadbang;
#X msg 356 268 read prism.normal;
#X obj 356 289 s \ $0-normal;
#X obj 337 197 t b b b;
15 #X msg 376 221 read prism.position;
#X obj 376 242 s \ $0-position;
#X obj 128 222 symbol \ $0-normal;
#X msg 128 243 normal \ $1;
#X obj 233 224 symbol \ $0-position;
20 #X msg 233 245 position \ $1;
#X obj 40 278 gemvertexbuffer 1536;
#X msg 160 157 draw triangle \, position_enable 1 \, normal_enable
1;
#X obj 298 40 t b a;
25 #X msg 298 88 link \ $1 \ $2;
#X msg 170 113 ambient 0.1 0.1 0.1 \, diffuse 0.7 0.7 0.7 \, specular
1 1 1 \, shine 16 \, light -10 10 10;
#X connect 0 0 1 0;
#X connect 1 0 2 0;
30 #X connect 1 1 21 0;
#X connect 2 0 3 0;
#X connect 2 1 4 0;
#X connect 3 0 19 0;
#X connect 4 0 22 0;
35 #X connect 5 0 6 0;
#X connect 5 0 23 0;

```

```

#X connect 6 0 3 0;
#X connect 9 0 12 0;
#X connect 10 0 11 0;
40 #X connect 12 0 20 0;
#X connect 12 1 17 0;
#X connect 12 1 15 0;
#X connect 12 2 13 0;
#X connect 12 2 10 0;
45 #X connect 13 0 14 0;
#X connect 15 0 16 0;
#X connect 16 0 19 0;
#X connect 17 0 18 0;
#X connect 18 0 19 0;
50 #X connect 20 0 19 0;
#X connect 21 0 4 0;
#X connect 21 1 4 1;
#X connect 22 0 3 0;
#X connect 23 0 3 0;

```

85 sketches/prism/prism.vert

```

#version 120

uniform float factor;

5   varying vec4 position;
    varying vec3 normal;

void main(void) {
    vec3 p = gl_Vertex.xyz;
10   vec3 n = gl_Normal.xyz;
    vec3 q = normalize(p);
    vec4 pos = gl_ModelViewProjectionMatrix * vec4(mix(p, q, factor), 1.0);
    normal = normalize(gl_NormalMatrix * vec3(normalize(mix(n, q, factor))));
    position = pos;
15   gl_Position = pos;
}

```

86 sketches/thunder/distance.pd

```

#N canvas 0 0 450 300 10;
#N canvas 0 0 205 392 distance 0;
#X obj 5 93 inlet~;
#X obj 4 370 outlet~;
5   #X obj 5 346 +~;
#X obj 32 199 delwrite~ distance 1000;
#X obj 21 243 sig~;
#X obj 5 115 lop~ 100;
#X obj 21 303 vd~ distance;
10  #X obj 21 323 *~ 0.5;
#X obj 67 2 inlet;
#X obj 87 29 swap 1;
#X obj 87 50 -;
#X obj 87 92 + 50;
15  #X obj 5 137 lop~ 100;
#X obj 5 159 lop~ 100;

```

```

#X obj 21 283 +~ 2;
#X obj 21 263 *~ 3;
#X obj 87 71 * 6000;
20 #X connect 0 0 5 0;
#X connect 2 0 1 0;
#X connect 4 0 15 0;
#X connect 5 0 12 0;
#X connect 6 0 7 0;
25 #X connect 7 0 2 1;
#X connect 8 0 4 0;
#X connect 8 0 9 0;
#X connect 9 0 10 0;
#X connect 9 1 10 1;
30 #X connect 10 0 16 0;
#X connect 11 0 5 1;
#X connect 11 0 12 1;
#X connect 11 0 13 1;
#X connect 12 0 13 0;
35 #X connect 13 0 2 0;
#X connect 13 0 3 0;
#X connect 14 0 6 0;
#X connect 15 0 14 0;
#X connect 16 0 11 0;
40 #X restore 71 128 pd distance;
#X obj 71 88 inlet ~;
#X obj 72 162 outlet ~;
#X obj 143 108 inlet distance;
#X connect 0 0 2 0;
45 #X connect 1 0 0 0;
#X connect 3 0 0 1;

```

87 sketches/thunder/strike-pattern.pd

```

#N canvas 0 0 450 300 10;
#X obj 47 9 inlet;
#X obj 47 30 t b b;
#X msg 47 51 1;
5 #X obj 47 72 +;
#X obj 47 93 moses 100;
#X obj 47 114 t f f;
#X obj 47 135 del;
#X obj 47 156 t b b;
10 #X obj 47 177 random 100;
#X obj 10 90 / 10;
#X obj 78 73 f;
#X msg 78 52 0;
#X obj 176 121 swap 100;
15 #X obj 176 142 -;
#X obj 176 163 / 100;
#X obj 110 154 f;
#X obj 136 144 + 1;
#X obj 136 165 mod 4;
20 #X obj 110 206 pack f f;
#X obj 110 227 outlet;
#X connect 0 0 1 0;
#X connect 1 0 2 0;
#X connect 1 1 11 0;

```

```

25  #X connect 2 0 3 0;
    #X connect 3 0 4 0;
    #X connect 3 0 10 0;
    #X connect 4 0 5 0;
    #X connect 5 0 6 0;
30  #X connect 5 1 12 0;
    #X connect 6 0 7 0;
    #X connect 7 0 8 0;
    #X connect 7 1 15 0;
    #X connect 8 0 9 0;
35  #X connect 9 0 3 0;
    #X connect 10 0 3 1;
    #X connect 11 0 10 0;
    #X connect 12 0 13 0;
    #X connect 12 1 13 1;
40  #X connect 13 0 14 0;
    #X connect 14 0 18 1;
    #X connect 15 0 16 0;
    #X connect 15 0 18 0;
    #X connect 16 0 17 0;
45  #X connect 17 0 15 1;
    #X connect 18 0 19 0;

```

88 sketches/thunder/strike-sound.pd

```

#N canvas 95 82 204 321 10;
#X obj 0 154 noise~;
#X obj 39 207 *~;
#X obj 55 175 vline~;
5  #X obj 136 96 + 100;
    #X obj 87 302 outlet~;
    #X obj 55 0 inlet;
    #X obj 39 265 bp~ 1 3;
    #X obj 144 266 bp~ 1 3;
10 #X obj 166 140 * 0.5;
    #X obj 55 45 swap 1;
    #X obj 55 68 -;
    #X msg 55 153 1 \, 0 \$1 0;
    #X obj 55 22 t f f;
15 #X obj 55 110 pow 5;
    #X obj 55 89 + 0.4;
    #X obj 136 117 t f f;
    #X obj 136 75 * 1200;
    #X obj 55 132 * 200;
20 #X connect 0 0 1 0;
    #X connect 1 0 6 0;
    #X connect 1 0 7 0;
    #X connect 2 0 1 1;
    #X connect 3 0 15 0;
25 #X connect 5 0 12 0;
    #X connect 6 0 4 0;
    #X connect 7 0 4 0;
    #X connect 8 0 7 1;
    #X connect 9 0 10 0;
30 #X connect 9 1 10 1;
    #X connect 10 0 14 0;
    #X connect 11 0 2 0;

```

```

#X connect 12 0 9 0;
#X connect 12 1 16 0;
35 #X connect 13 0 17 0;
#X connect 14 0 13 0;
#X connect 15 0 6 1;
#X connect 15 1 8 0;
#X connect 16 0 3 0;
40 #X connect 17 0 11 0;

```

89 sketches/thunder/thunder4.pd

```

#N canvas 1302 0 601 433 10;
#X obj 248 113 *~ 0.9;
#N canvas 152 29 188 323 afterimage 0;
#X obj 1 114 noise~;
5 #X obj 16 176 noise~;
#X obj 1 199 *~;
#X obj 1 241 bp~ 333 4;
#X obj 1 135 lop~ 3;
#X obj 1 220 clip~ -1 1;
10 #X obj 52 68 line;
#X obj 52 109 + 3;
#X obj 52 89 * 30;
#X obj 1 156 *~ 80;
#X obj 117 67 line;
15 #X obj 1 263 *~;
#X msg 52 48 1 \, 0 600;
#X msg 117 47 1 \, 0 6000;
#X obj 1 283 *~ 0.9;
#X obj 85 23 del 200;
20 #X obj 85 1 inlet;
#X obj 1 303 outlet~;
#X connect 0 0 4 0;
#X connect 1 0 2 1;
#X connect 2 0 5 0;
25 #X connect 3 0 11 0;
#X connect 4 0 9 0;
#X connect 5 0 3 0;
#X connect 6 0 8 0;
#X connect 7 0 4 1;
30 #X connect 8 0 7 0;
#X connect 9 0 2 0;
#X connect 10 0 11 1;
#X connect 11 0 14 0;
#X connect 12 0 6 0;
35 #X connect 13 0 10 0;
#X connect 14 0 17 0;
#X connect 15 0 12 0;
#X connect 15 0 13 0;
#X connect 16 0 15 0;
40 #X restore 248 91 pd afterimage;
#X obj 272 262 distance;
#X obj 323 238 line;
#X obj 11 71 r go;
#X obj 248 68 r go;
45 #X obj 350 66 r go;
#X obj 128 70 r go;

```

```

#X obj 186 49 line;
#X msg 186 26 1 \, 0 9000;
#X obj 186 4 r go;
50 #X obj 323 193 r go;
#X obj 272 348 dac~;
#X obj 286 288 s~ udly;
#N canvas 265 516 402 108 box 0;
#X obj 319 51 switch~;
55 #X obj 319 29 inlet;
#X obj 1 7 udly 12000;
#X obj 11 17 udly 12000;
#X obj 21 27 udly 12000;
#X obj 31 37 udly 12000;
60 #X obj 41 47 udly 12000;
#X obj 51 57 udly 12000;
#X obj 61 67 udly 12000;
#X obj 71 77 udly 12000;
#X obj 87 7 udly 12000;
65 #X obj 97 17 udly 12000;
#X obj 107 27 udly 12000;
#X obj 117 37 udly 12000;
#X obj 127 47 udly 12000;
#X obj 137 57 udly 12000;
70 #X obj 147 67 udly 12000;
#X obj 157 77 udly 12000;
#X obj 172 6 udly 12000;
#X obj 182 16 udly 12000;
#X obj 192 26 udly 12000;
75 #X obj 202 36 udly 12000;
#X obj 212 46 udly 12000;
#X obj 222 56 udly 12000;
#X obj 232 66 udly 12000;
#X obj 242 76 udly 12000;
80 #X connect 1 0 0 0;
#X restore 465 135 pd box of delays;
#X obj 126 264 catch~ la;
#X obj 368 263 catch~ ra;
#X obj 465 115 tgl 15 0 empty empty empty 0 -6 0 8 -262144 -1 -1 1
85 1;
#X obj 490 112 s shake;
#X obj 490 91 r go;
#X obj 12 12 bng 15 250 50 0 empty empty empty 0 -6 0 8 -262144 -1
-1;
90 #X obj 12 34 s go;
#X obj 186 71 * 2000;
#X msg 323 215 0 \, 1 12000;
#X obj 248 135 throw~ out;
#X obj 197 264 catch~ out;
95 #X obj 350 135 throw~ distance;
#X obj 129 135 throw~ distance;
#X obj 272 173 catch~ distance;
#N canvas 132 138 177 510 rumble 0;
#X obj 17 110 noise~;
100 #X obj 33 314 ~ 0.5;
#X obj 33 339 *~;
#X obj 33 359 *~ -4;
#X obj 33 379 +~ 1;

```

```
105 #X obj 33 399 *~ 0.5;
#X obj 75 28 noise~;
#X obj 75 89 max~ 0;
#X obj 17 172 samphold~;
#X msg 938 102 0.54;
#X obj 75 151 phasor~;
110 #X obj 75 111 *~ 3000;
#X obj 17 419 *~;
#X obj 17 439 hip~ 100;
#X obj 17 192 rpole~ 0.99;
#X obj 17 212 *~ 0.04;
115 #X obj 75 131 +~ 1;
#X obj 81 -12 inlet density;
#X obj 17 131 lop~;
#X obj 17 152 lop~;
#X obj 75 68 lop~;
120 #X obj 75 48 lop~;
#X obj 1 480 outlet~;
#X obj 17 241 min~ 0;
#X obj 17 263 *~ -1;
#X obj 65 240 max~ 0;
125 #X obj 17 286 +~;
#X obj 1 28 vline~;
#X obj 1 459 *~;
#X obj 1 -12 inlet trig;
#X msg 1 8 1 50 \, 0 8000 50;
130 #X connect 0 0 18 0;
#X connect 1 0 2 0;
#X connect 1 0 2 1;
#X connect 2 0 3 0;
#X connect 3 0 4 0;
135 #X connect 4 0 5 0;
#X connect 5 0 12 1;
#X connect 6 0 21 0;
#X connect 7 0 11 0;
#X connect 8 0 14 0;
140 #X connect 10 0 8 1;
#X connect 11 0 16 0;
#X connect 12 0 13 0;
#X connect 13 0 28 1;
#X connect 14 0 15 0;
145 #X connect 15 0 25 0;
#X connect 15 0 23 0;
#X connect 16 0 10 0;
#X connect 17 0 20 1;
#X connect 17 0 21 1;
150 #X connect 17 0 18 1;
#X connect 17 0 19 1;
#X connect 18 0 19 0;
#X connect 19 0 8 0;
#X connect 20 0 7 0;
155 #X connect 21 0 20 0;
#X connect 23 0 24 0;
#X connect 24 0 26 0;
#X connect 25 0 26 1;
#X connect 26 0 1 0;
160 #X connect 26 0 12 0;
```

```
#X connect 27 0 28 0;
#X connect 28 0 22 0;
#X connect 29 0 30 0;
#X connect 30 0 27 0;
165 #X restore 128 93 pd rumble;
#N canvas 264 575 152 190 strike 0;
#X obj 0 172 outlet~;
#X obj 1 44 route 0 1 2 3;
#X obj 1 0 inlet;
170 #X obj 1 67 strike-sound;
#X obj 22 87 strike-sound;
#X obj 44 107 strike-sound;
#X obj 65 126 strike-sound;
#X obj 1 23 strike-pattern;
175 #X connect 1 0 3 0;
#X connect 1 1 4 0;
#X connect 1 2 5 0;
#X connect 1 3 6 0;
#X connect 2 0 7 0;
180 #X connect 3 0 0 0;
#X connect 4 0 0 0;
#X connect 5 0 0 0;
#X connect 6 0 0 0;
#X connect 7 0 1 0;
185 #X restore 11 94 pd strike sound;
#N canvas 148 116 232 212 deep 0;
#X obj 8 -8 noise~;
#X obj 7 161 *~;
#X obj 84 115 *~;
190 #X obj 84 49 del 1000;
#X obj 84 28 inlet;
#X obj 7 181 outlet~;
#X obj 84 91 vline~;
#X msg 84 70 1 2000 \, 0 20000 2000;
195 #X obj 8 12 lop~ 80;
#X obj 8 32 lop~ 80;
#X obj 8 52 hip~ 15;
#X obj 8 92 clip~ -1 1;
#X obj 8 72 *~ 12;
200 #X obj 8 112 lop~ 80;
#X obj 8 132 lop~ 80;
#X connect 0 0 8 0;
#X connect 1 0 5 0;
#X connect 2 0 1 1;
205 #X connect 3 0 7 0;
#X connect 4 0 3 0;
#X connect 6 0 2 0;
#X connect 6 0 2 1;
#X connect 7 0 6 0;
210 #X connect 8 0 9 0;
#X connect 9 0 10 0;
#X connect 10 0 12 0;
#X connect 11 0 13 0;
#X connect 12 0 11 0;
215 #X connect 13 0 14 0;
#X connect 14 0 1 0;
#X restore 350 88 pd deep;
```



```

#X obj 11 135 throw~ out;
#X obj 11 115 *~ 0.3;
220 #X obj 129 114 *~ 0.3;
#X obj 350 111 *~ 0.5;
#X obj 364 342 r rec;
#X obj 364 363 sel 1 0;
#X obj 222 369 writesf~ 2;
225 #X msg 385 386 stop;
#X msg 310 416 open -bytes 4 o.wav \, start;
#X obj 467 39 loadbang;
#X msg 467 60 1;
#X connect 0 0 24 0;
230 #X connect 1 0 0 0;
#X connect 2 0 12 0;
#X connect 2 0 12 1;
#X connect 2 0 13 0;
#X connect 2 0 38 0;
235 #X connect 2 0 38 1;
#X connect 3 0 2 1;
#X connect 4 0 30 0;
#X connect 5 0 1 0;
#X connect 6 0 31 0;
240 #X connect 7 0 29 0;
#X connect 8 0 22 0;
#X connect 9 0 8 0;
#X connect 10 0 9 0;
#X connect 11 0 23 0;
245 #X connect 15 0 12 0;
#X connect 15 0 38 0;
#X connect 16 0 12 1;
#X connect 16 0 38 1;
#X connect 17 0 14 0;
250 #X connect 19 0 18 0;
#X connect 20 0 21 0;
#X connect 22 0 29 1;
#X connect 23 0 3 0;
#X connect 25 0 12 0;
255 #X connect 25 0 12 1;
#X connect 25 0 38 0;
#X connect 25 0 38 1;
#X connect 28 0 2 0;
#X connect 29 0 34 0;
260 #X connect 30 0 33 0;
#X connect 31 0 35 0;
#X connect 33 0 32 0;
#X connect 34 0 27 0;
#X connect 35 0 26 0;
265 #X connect 36 0 37 0;
#X connect 37 0 40 0;
#X connect 37 1 39 0;
#X connect 39 0 38 0;
#X connect 40 0 38 0;
270 #X connect 41 0 42 0;
#X connect 42 0 17 0;

```

90 sketches/thunder/udly.pd

```

#N canvas 128 91 234 472 10;
#X obj 82 24 delwrite~ \$0-a \$1;
#X obj 5 234 delread~ \$0-a;
#X obj 82 1 r~ udly;
5 #X obj 5 1 r shake;
#X obj 5 387 *~;
#X obj 20 163 \$1;
#X obj 20 189 -;
#X obj 20 211 / \$1;
10 #X obj 5 22 t b b;
#X obj 5 89 random;
#X obj 35 45 \$1;
#X obj 35 66 * 10000;
#X obj 5 109 / 10000;
15 #N canvas 0 0 450 300 cpan 0;
#X obj -2 -2 inlet~ signal;
#X obj 88 192 *~;
#X obj -2 192 *~;
#X obj 104 110 cos~;
20 #X obj 104 87 -~ 0.25;
#X obj 104 65 *~ 0.25;
#X obj -2 220 outlet~ left;
#X obj 88 220 outlet~ right;
#X obj 14 166 cos~;
25 #X obj 14 142 -~ 0.25;
#X obj 107 37 inlet~;
#X connect 0 0 2 0;
#X connect 0 0 1 0;
#X connect 1 0 7 0;
30 #X connect 2 0 6 0;
#X connect 3 0 1 1;
#X connect 4 0 3 0;
#X connect 4 0 9 0;
#X connect 5 0 4 0;
35 #X connect 8 0 2 1;
#X connect 9 0 8 0;
#X connect 10 0 5 0;
#X restore 5 410 pd cpan;
#X obj 5 449 throw~ la;
40 #X obj 75 448 throw~ ra;
#X obj 59 160 random 10000;
#X obj 21 267 *;
#X obj 5 130 t f b f b;
#X obj 59 183 / 20000;
45 #X obj 59 206 + 0.5;
#X obj 36 315 * 400;
#X obj 36 337 + 400;
#X obj 5 364 bp~ 1000 2;
#X connect 1 0 23 0;
50 #X connect 2 0 0 0;
#X connect 3 0 8 0;
#X connect 4 0 13 0;
#X connect 5 0 6 0;
#X connect 6 0 7 0;
55 #X connect 7 0 17 0;
#X connect 8 0 9 0;
#X connect 8 1 10 0;

```

```
#X connect 9 0 12 0;
#X connect 10 0 11 0;
60 #X connect 11 0 9 1;
#X connect 12 0 18 0;
#X connect 13 0 14 0;
#X connect 13 1 15 0;
#X connect 16 0 19 0;
65 #X connect 17 0 4 1;
#X connect 17 0 13 1;
#X connect 17 0 21 0;
#X connect 18 0 1 0;
#X connect 18 1 5 0;
70 #X connect 18 2 6 1;
#X connect 18 3 16 0;
#X connect 19 0 20 0;
#X connect 20 0 17 1;
#X connect 21 0 22 0;
75 #X connect 22 0 23 1;
#X connect 23 0 4 0;
```