

```

0 ( FRACT-OV ----- OVERLAY PREFIX -----
1 ( FRACT-OV  CONSTANTS & VARIABLES
2 ( FRACT-OV  CONSTANTS & VARIABLES
3 ( FRACT-OV  SETSCALE GENANCHORS
4 ( FRACT-OV  [RRND] C+LIMIT
5 ( FRACT-OV  DISPLACMNT
6 ( FRACT-OV  [SWRAP]
7 ( FRACT-OV  [ACELLADDR]
8 ( FRACT-OV  [A@]
9 ( FRACT-OV  !HEIGHT [!HEIGHT]
10 ( FRACT-OV  SETANCHORS
11 ( FRACT-OV  XSHIFT
12 ( FRACT-OV  YSHIFT
13 ( FRACT-OV  EDGES
14 ( FRACT-OV  CENTER
15 ( FRACT-OV  MIDPT NEWSTD
16 ( FRACT-OV  [FRACTAL]
17 ( FRACT-OV  [FRACTAL] con't
18 ( FRACT-OV  FRACTALIZE FILLARRAY
19 ( FRACT-OV  FRACT-REGION
20 ( FRACT-OV  MERCATOR-GEN
21 ( FRACT-OV  CONTOUR-RATIO CONTOUR
22 ( FRACT-OV  INIT-CONTOUR
23 ( FRACT-OV  MERC>CONANCHOR
24 ( FRACT-OV  AV-MIDPT
25 ( FRACT-OV  CONANCHOR-HOR
26 ( FRACT-OV  CONANCHOR-VER CONANCHOR-CNT1
27 ( FRACT-OV  CONANCHOR-CNT2
28 ( FRACT-OV  CONANCHOR>CONTOUR
29 ( FRACT-OV  ANCHOR-CONTOUR
30 ( FRACT-OV  SUB-CON-FRACT
31 ( FRACT-OV  FRACT-CONTOUR
32 ( FRACT-OV  +FRAME -FRAME @FRAME !FRAME
33 ( FRACT-OV  @X1 @X2 @XMID @Y1 @Y2 @YMID !XMID !YMI
34 ( FRACT-OV - OUTPOLY
35 ( FRACT-OV - ?HOMOGENOUS
36 ( FRACT-OV - POLYGON-EXTRACT
37 ( FRACT-OV - BUILD-CONTINENTS
38 ( FRACT-OV - +TMP -TMP @TMP !TMP
39 ( FRACT-OV - CACCUM
40 ( FRACT-OV - MAJOR-COLOR
41 ( FRACT-OV - DROP-POLY PUSH-POLY POP-POLY
42 ( FRACT-OV - ?MCLR MAINPOLY
43 ( FRACT-OV - REDUCE-FACE
44 ( FRACT-OV - REDUCE-ALL MAKE-GLOBE
45 ( FRACT-OV - UPSCRL DNSCRL
46 ( FRACT-OV - RTSCRL LFSCRL
47 ( FRACT-OV - SCRLXCON
48 ( FRACT-OV - SCRLYCON
49 ( FRACT-OV - SCRLCON
50 ( FRACT-OV ----- OVERLAY SUFFIX -----
51
52
53

```

0

3

```

0 ( FRACT-OV ----- OVERLAY PREFIX ----- 8-13-85) ( FRACT-OV SETSCALE GENANCHORS 11-19-84)
1
2 VOCABULARY FRACT IMMEDIATE HEAD: SETSCALE ( dratio scale -- \ set )
3 ( deviation change ratio and scale)
4 71 OPEN-OVERLAY T: STD ! RATIO 2! T;
5
6 FRACT DEFINITIONS HEAD: GENANCHORS ( -- All Aul Aur Alr \ )
7 ( generate 4 anchor points)
8 2500 TRANS-ALLOT NEWT-DP T: 4 0 DO STD @ DUP NEGATE SWAP RRND
9 LOOP T;
10
11
12
13
14
15

```

1

4

```

0 ( FRACT-OV CONSTANTS & VARIABLES 8-05-85) ( FRACT-OV [RRND] C+LIMIT 8-12-85)
1 V: XYANCHOR ( x,y abs anchor flag)
2 V: STD ( standard deviation range)
3 2V: RATIO ( internal roughness control)
4 -128 C: FNULL ( fractal null value)
5
6 V: DY>1 ( delta y is > 1 flag)
7 V: DX>1 ( delta x is > 1 flag)
8 V: XO' ( translated origin - abs)
9 V: YO' ( translated origin - abs)
10 V: SCELL ( seq address for current cell)
11 V: OCELL ( off address for current cell)
12 20882 32767 2C: MERCATOR-RATIO
13 117 C: MERCATOR-SCALE
14
15

```

```

HEX
V: [RRND] -2 DP +!
( lo hi rtn -- n \ generate a random # in the )
( range lo to hi)
8B C, 06 C, SEED , 89 C, BD C, 7A C, F7 C, E9 C, 05 C, 0F C,
1B C, 89 C, 06 C, SEED , 5A C, 5B C, 59 C, 52 C, 2B C, D9 C,
F7 C, E3 C, 03 C, D1 C, 5B C, 52 C, 50 C, C3 C,

V: C+LIMIT -2 DP +!
( c c rtn -- c \ add 2 bytes, clipping to limits)
5B C, 5B C, 59 C, 03 C, C1 C, 3D C, 7F C, 00 C, 7E C, 03 C,
8B C, 7F C, 00 C, 3D C, 81 C, FF C, 79 C, 03 C, 8B C, 81 C,
FF C, 50 C, 53 C, C3 C,

DECIMAL

```

2

5

```

0 ( FRACT-OV CONSTANTS & VARIABLES 8-05-85) ( FRACT-OV DISPLACMNT 8-14-85)
1
2 V: RTO ( return addr holder)
3 V: RTI ( return addr holder)
4 V: RTEMP ( return stack pointer holder)
5 V: TY ( temp variable for xshift, yshift)
6 EXIT
7 2 C= FY2 ( frame offset for y2)
8 4 C= FX2 ( frame offset for x2)
9 6 C= FY1 ( frame offset for y1)
10 8 C= FX1 ( frame offset for x1)
11 10 C= FYMID ( frame offset for ymid)
12 12 C= FXMID ( frame offset for xmid)
13 14 C= FSTD ( frame offset for std)
14
15

```

```

HEX V: DISPLACMNT -2 DP +! ( xrel yrel rtn -- n \ gen a)
( fractal displacement given relative location w/ optional)
( fixed seeds to make fractal stationary)
5B C, 5B C, 59 C, 53 C, 8B C, 16 C, XYANCHOR , 0B C, D2 C,
7E C, 50 C, 03 C, 0E C, XO' , 03 C, 06 C, YO' , 79 C, 09 C,
40 C, F7 C, D8 C, 81 C, C1 C, 80 C, 04 C, EB C, 12 C, BA C,
C0 C, 03 C, 3B C, C2 C, 7B C, 0B C, 2B C, C2 C, F7 C, D8 C,
03 C, C2 C, 40 C, 81 C, C1 C, 80 C, 04 C, 89 C, 06 C, SEED ,
51 C, 8B C, 01 C, 00 C, 50 C, F7 C, D8 C, 50 C, EB C,
[RRND] HERE 2+ - , 59 C, 8F C, 06 C, SEED , 51 C, 8B C,
01 C, 00 C, 50 C, F7 C, D8 C, 50 C, EB C, [RRND]
HERE 2+ - , 5B C, 59 C, 33 C, 06 C, GLOBALSEED , 33 C, C1 C,
89 C, 06 C, SEED , 8B C, 46 C, 0E C, 8B C, C8 C, F7 C, D9 C,
51 C, 50 C, EB C, [RRND] HERE 2+ - , 5B C, 59 C, 50 C, 51 C,
C3 C, DECIMAL

```

6

```

0 ( FRACT-OV [SWRAP]
1 HEX
2 V: [SWRAP] -2 DP +!
3 ( x y rtn -- x'y' \ calc new array cell coords)
4 ( using spherical wrap)
5 5A C, 58 C, 59 C, 0B C, C0 C, 79 C, 0F C, 40 C, F7 C, D8 C,
6 8B C, 1E C, 'ARRAY, 8B C, 1F C, D1 C, EB C, 03 C, CB C,
7 EB C, 1C C, 8B C, 1E C, 'ARRAY, 8B C, 5F C, 02 C, 3B C,
8 C3 C, 78 C, 11 C, 2B C, C3 C, F7 C, D8 C, 03 C, C3 C, 40 C,
9 8B C, 1E C, 'ARRAY, 8B C, 1F C, D1 C, EB C, 03 C, CB C,
10 0B C, C9 C, 79 C, 0B C, 8B C, 1E C, 'ARRAY, 03 C, 0F C,
11 EB C, 0C C, 8B C, 1E C, 'ARRAY, 8B C, 1F C, 3B C, CB C,
12 78 C, 02 C, 2B C, CB C, 51 C, 50 C, 52 C, C3 C,
13 DECIMAL
14
15

```

7

```

0 ( FRACT-OV [ACELLADDR]
1 HEX
2 V: [ACELLADDR] -2 DP +! ( x y rtn -- \ compute array )
3 ( cell address with optional spherical boundary wrapping)
4 8F C, 06 C, RT0 , 8B C, 06 C, SPHEREWAP , 0B C, C0 C,
5 7E C, 03 C, EB C, [SWRAP] HERE 2+ - , 8B C, 1E C, 'ARRAY ,
6 8B C, 47 C, 06 C, 8B C, 5F C, 04 C, 59 C, D1 C, E1 C, 03 C,
7 D9 C, 1E C, 8E C, D8 C, 8B C, 0F C, 1F C, 5A C, 03 C, CA C,
8 89 C, 06 C, SCELL , 89 C, 0E C, OCELL , FF C, 36 C,
9 RT0 , C3 C,
10 DECIMAL
11
12
13
14
15

```

8

```

0 ( FRACT-OV [A@]
1
2 HEX
3 V: [A@] -2 DP +! ( rtn -- c or n \ long byte fetch with )
4 ( optional sign extension for current array cell)
5 5A C, 8B C, 1E C, OCELL , 1E C, FF C, 36 C, SCELL ,
6 1F C, 8A C, 07 C, 1F C, 8B C, 0E C, SIGNEXTEND ,
7 0B C, C9 C, 7E C, 03 C, 98 C, EB C, 02 C, 32 C, E4 C,
8 50 C, 52 C, C3 C,
9 DECIMAL
10
11
12
13
14
15

```

9

```

8-12-85) ( FRACT-OV !HEIGHT [!HEIGHT] 8-12-85)
HEX
V: [!HEIGHT] -2 DP +!
( val x y rtn -- \ set height in current array)
( if value of cell xy is null)
8F C, 06 C, RT1 , EB C, [ACELLADDR] HERE 2+ - , EB C, [A@]
HERE 2+ - , 58 C, 3D C, FNULL , 75 C, 10 C, 58 C, 1E C,
FF C, 36 C, SCELL , 8B C, 1E C, OCELL , 1F C, 8B C, 07 C,
1F C, EB C, 01 C, 58 C, FF C, 36 C, RT1 , C3 C,
V: !HEIGHT -2 DP +! ( val x y -- \ see [!height]) HERE DUP 2- !
EB C, [!HEIGHT] HERE 2+ - , AD C, 8B C, D8 C, FF C, 27 C,
DECIMAL

```

10

```

8-12-85) ( FRACT-OV SETANCHORS 8-06-85)
HEAD: SETANCHORS
( All Aul Aur Alr \ set anchor pts for region.)
T: XUR @ YLL @ !HEIGHT
XUR @ YUR @ !HEIGHT
XLL @ YUR @ !HEIGHT
XLL @ YLL @ !HEIGHT T;

```

11

```

8-12-85) ( FRACT-OV XSHIFT 8-12-85)
HEX
V: XSHIFT -2 DP +! ( y rtn -- \ displace midpt of x1,y x2,y)
59 C, 8F C, 06 C, TY , 51 C, FF C, 76 C, 0B C, FF C, 36 C,
TY , EB C, [ACELLADDR] HERE 2+ - , EB C, [A@] HERE 2+ - ,
FF C, 76 C, 04 C, FF C, 36 C, TY , EB C, [ACELLADDR] HERE
2+ - , EB C, [A@] HERE 2+ - , 58 C, 59 C, 03 C, C1 C,
D1 C, FB C, 50 C, 8B C, 5E C, 0B C, 03 C, 5E C, 04 C, D1 C,
FB C, 53 C, FF C, 36 C, TY , EB C, DISPLACMNT HERE 2+ - ,
EB C, C+LIMIT HERE 2+ - , FF C, 76 C, 0C C, FF C, 36 C, TY ,
EB C, [!HEIGHT] HERE 2+ - , C3 C,
DECIMAL

```


12

```

0 ( FRACT-OV  YSHIFT
1 HEX
2 V: YSHIFT -2 DP +! ( x rtn -- \ displace midpt of x,y1 x,y2)
3 59 C, 8F C, 06 C, TY , 51 C, FF C, 36 C, TY , FF C, 76 C,
4 06 C, E8 C, [ACELLADDR] HERE 2+ - , E8 C, [A@] HERE 2+ - ,
5 FF C, 36 C, TY , FF C, 76 C, 02 C, E8 C, [ACELLADDR] HERE
6 2+ - , E8 C, [A@] HERE 2+ - , 58 C, 59 C, 03 C, C1 C,
7 D1 C, F8 C, 50 C, FF C, 36 C, TY , 8B C, 5E C, 06 C,
8 03 C, 5E C, 02 C, D1 C, F8 C, 53 C, E8 C, DISPLACMNT HERE
9 2+ - , E8 C, C+LIMIT HERE 2+ - , FF C, 36 C, TY , FF C,
10 76 C, 0A C, E8 C, [!HEIGHT] HERE 2+ - , C3 C,
11 DECIMAL
12
13
14
15

```

13

```

0 ( FRACT-OV  EDGES
1 HEX
2 V: EDGES -2 DP +! ( -- \ compute heights for edge midpoints)
3 8B C, 06 C, DY>1 , 0B C, C0 C, 7E C, 11 C, 8B C, 46 C, 0B C,
4 0B C, C0 C, 75 C, 04 C, 50 C, E8 C, YSHIFT HERE 2+ - ,
5 FF C, 76 C, 04 C, E8 C, YSHIFT HERE 2+ - , 8B C, 06 C, DX>1 ,
6 0B C, C0 C, 7E C, 11 C, 8B C, 46 C, 06 C, 0B C, C0 C, 75 C,
7 04 C, 50 C, E8 C, XSHIFT HERE 2+ - , FF C, 76 C, 02 C,
8 E8 C, XSHIFT HERE 2+ - , C3 C,
9 DECIMAL
10
11
12
13
14
15

```

14

```

0 ( FRACT-OV  CENTER
1 HEX
2 V: CENTER -2 DP +! ( -- \ displace center of current rectangle)
3 8B C, 06 C, DY>1 , 23 C, 06 C, DX>1 , 7E C, 54 C, FF C,
4 76 C, 0C C, FF C, 76 C, 06 C, E8 C, [ACELLADDR] HERE 2+ - ,
5 E8 C, [A@] HERE 2+ - , FF C, 76 C, 0C C, FF C, 76 C, 02 C,
6 E8 C, [ACELLADDR] HERE 2+ - , E8 C, [A@] HERE 2+ - , FF C,
7 76 C, 0B C, FF C, 76 C, 0A C, E8 C, [ACELLADDR] HERE 2+ - ,
8 E8 C, [A@] HERE 2+ - , FF C, 76 C, 04 C, FF C, 76 C, 0A C,
9 E8 C, [ACELLADDR] HERE 2+ - , E8 C, [A@] HERE 2+ - , 58 C,
10 59 C, 03 C, C1 C, 59 C, 03 C, C1 C, 59 C, 03 C, C1 C, D1 C,
11 F8 C, D1 C, F8 C, 50 C, FF C, 76 C, 0C C, FF C, 76 C, 0A C,
12 E8 C, DISPLACMNT HERE 2+ - , E8 C, C+LIMIT HERE 2+ - ,
13 FF C, 76 C, 0C C, FF C, 76 C, 0A C, E8 C, [!HEIGHT] HERE 2+
14 - , C3 C,
15 DECIMAL

```

15

```

8-12-85) ( FRACT-OV  MIDPT NEWSTD
HEX
V: MIDPT -2 DP +! ( compute midpoint)
8B C, 46 C, 0B C, 03 C, 46 C, 04 C, D1 C, F8 C, 89 C, 46 C,
0C C, 8B C, 46 C, 06 C, 03 C, 46 C, 02 C, D1 C, F8 C, 89 C,
46 C, 0A C, C3 C,

V: NEWSTD -2 DP +! ( rtn -- \ compute new standard deviation)
8B C, 46 C, 0E C, F7 C, 26 C, RATIO 2+ , F7 C, 36 C, RATIO ,
0B C, C0 C, 7F C, 03 C, 8B C, 01 C, 00 C, 89 C, 46 C, 0E C,
C3 C,

DECIMAL

```

16

```

8-12-85) ( FRACT-OV  [FRACTAL]
HEX V: [FRACTAL] -2 DP +! ( std xmid ymid x1 y1 x2 y2 rtn -- )
8B C, EC C, 8B C, 46 C, 04 C, 2B C, 46 C, 0B C, 48 C, 7E C,
05 C, 8B C, 01 C, 00 C, E8 C, 02 C, 33 C, C0 C, 89 C, 06 C,
DX>1 , 8B C, 4E C, 02 C, 2B C, 4E C, 06 C, 49 C, 7E C, 05 C,
89 C, 01 C, 00 C, E8 C, 02 C, 33 C, C9 C, 89 C, 0E C, DY>1 ,
0B C, C1 C, 7E C, 64 C, E8 C, MIDPT HERE 2+ - , E8 C, EDGES
HERE 2+ - , E8 C, CENTER HERE 2+ - , E8 C, NEWSTD HERE 2+ - ,
33 C, C0 C, FF C, 76 C, 0E C, 50 C, 50 C, FF C, 76 C, 0B C,
FF C, 76 C, 06 C, FF C, 76 C, 0C C, FF C, 76 C, 0A C, E8 C,
[FRACTAL] HERE 2+ - , 33 C, C0 C, FF C, 76 C, 0E C, 50 C,
50 C, FF C, 76 C, 0C C, FF C, 76 C, 06 C, FF C, 76 C, 04 C,
FF C, 76 C, 0A C, E8 C, [FRACTAL] HERE 2+ - , 33 C, C0 C,
FF C, 76 C, 0E C, 50 C, 50 C, FF C, 76 C, 0B C, FF C, 76 C,
0A C, FF C, 76 C, 0C C, FF C, 76 C, 02 C, E8 C, [FRACTAL]
HERE 2+ - , 33 C, C0 C, FF C, 76 C, 0E C, 50 C, 50 C,

```

17

```

8-13-85) ( FRACT-OV  [FRACTAL] con't
HEX
FF C, 76 C, 0C C, FF C, 76 C, 0A C, FF C, 76 C, 04 C,
FF C, 76 C, 02 C, E8 C, [FRACTAL] HERE 2+ - , 58 C, 83 C,
C4 C, 0E C, 8B C, EC C, 50 C, C3 C,
DECIMAL

```

18

21

```

0 ( FRACT-OV  FRACTALIZE  FILLARRAY
1 HEX
2 V: FRACTALIZE -2 DP +!
3 ( std 0 0 x1 y1 x2 y2 --) HERE DUP 2- !
4 89 C, 2E C, RTEMP , E8 C, [FRACTAL] HERE 2+ - , 8B C,
5 2E C, RTEMP , AD C, 8B C, DB C, FF C, 27 C,
6
7 V: FILLARRAY -2 DP +! ( val -- \ fill current array with val)
8 HERE DUP 2- ! 5B C, 8B C, 1E C, 'ARRAY , 8B C, 4F C, 04 C,
9 8B C, 5F C, 06 C, 06 C, 57 C, 8E C, C3 C, 33 C, FF C, F3 C,
10 AA C, 5F C, 07 C, AD C, 8B C, DB C, FF C, 27 C,
11 DECIMAL
12
13
14
15

```

```

8-13-85) ( FRACT-OV  CONTOUR-RATIO CONTOUR
20882 32767 2C: CONTOUR-RATIO
20 C: CONTOUR-SCALE ( STD)

```

11-19-84)

19

22

```

0 ( FRACT-OV  FRACT-REGION
1
2 HEAD: FRACT-REGION ( -- \ do a fractal for)
3 ( current region)
4 T: STD @ 0 0 XLL @ YLL @
5 XUR @ YUR @ FRACTALIZE T;
6
7
8
9
10
11
12
13
14
15

```

```

11-19-84) ( FRACT-OV  INIT-CONTOUR
: INIT-CONTOUR ( -- \ initialize the)
( contour array to null.)
' CONTOUR ( DUP !OFFSETS) SETLARRAY
( FULLARRAY SETREGION)
FNULL FILLARRAY ;

```

8-13-85)

20

23

```

0 ( FRACT-OV  MERCATOR-GEN
1
2 : MERCATOR-GEN ( seed -- \ generate a)
3 ( fractal mercator map given seed)
4 DUP SEED ! GLOBALSEED !
5 XYANCHOR OFF
6 MERCATOR-RATIO MERCATOR-SCALE
7 SETSCALE
8 SPHEREWRAP ON SIGNEXTEND ON
9 ' MERCATOR SETLARRAY
10 0 0 48 23 SETREGION
11 FNULL FILLARRAY
12 GENANCHORS
13 SETANCHORS
14 FRACT-REGION ;
15

```

```

8-13-85) ( FRACT-OV  MERC>CONANCHOR
HEAD: MERC>CONANCHOR ( -- \ populate )
( conachor from mercator array)
( ' CONANCHOR !OFFSETS)
T: YCON @ 40 / Y2 !
XCON @ 48 / X2 !
SPHEREWRAP ON ' MERCATOR SETLARRAY
4 0 DO ( 4 rows)
3 0 DO ( 3 cols)
I X2 @ + Y2 @ J + ACCELLADOR A@
I 4 * J 2* CONANCHOR LC!
LOOP LOOP T;

```

8-13-85)

24

```

0 ( FRACT-OV AV-MIDPT
1 HEAD: AV-MIDPT ( x1 y1 x2 y2 -- \ )
2 ( find midpt and store average val)
3 T: 2OVER ACELLADDR A@ >R ( -- n1)
4 2DUP ACELLADDR A@ >R + 2/ >R ( -- n')
5 ROT + 2/ >R ( x1 x2 -- y' n')
6 + 2/ >R >R ( x' y' n' --)
7 ROT ROT ( n x' y' --)
8 ACELLADDR A! T;
9
10
11
12
13
14
15

```

25

```

0 ( FRACT-OV CONANCHOR-HOR
1 HEAD: CONANCHOR-HOR ( -- \ average )
2 ( horizontals to fill in anchors)
3 T: 4 0 DO
4 2 0 DO
5 I 4 * J 2* OVER 4 + OVER AV-MIDPT
6 I 4 * J 2* OVER 2 + OVER AV-MIDPT
7 I 1+ 4 * J 2* OVER 2 - OVER AV-MIDPT
8 LOOP LOOP T;
9
10
11
12
13
14
15

```

26

```

0 ( FRACT-OV CONANCHOR-VER CONANCHOR-CNT1
1 HEAD: CONANCHOR-VER ( -- \ average )
2 ( verticals to fill in anchors)
3 T: 3 0 DO
4 3 0 DO
5 I 4 * J 2* OVER OVER 2+ AV-MIDPT
6 LOOP LOOP T;
7
8 HEAD: CONANCHOR-CNT1 ( -- \ average )
9 ( to fill center holes)
10 T: 3 0 DO
11 2 0 DO
12 I 4 * J 2* 1+ OVER 4 + OVER AV-MIDPT
13 LOOP LOOP T;
14
15

```

27

```

11-19-84) ( FRACT-OV CONANCHOR-CNT2
HEAD: CONANCHOR-CNT2 ( -- \ average )
( to fill center holes)
T: 3 0 DO
4 0 DO
I 2* 1+ J 2* OVER OVER 2+ AV-MIDPT
LOOP LOOP T;

HEAD: SETRELOIGIN
T: YD' ! XD' ! T;

```

11-19-84)

28

```

11-19-84) ( FRACT-OV CONANCHOR)CONTOUR
HEAD: CONANCHOR)CONTOUR ( -- \ move anchors)
( from conanchor to the contour array)
T: XCON @ 48 MOD 12 /
YCON @ 40 MOD 20 / ( x y --origin)
( in conanchor)
OVER XCON @ 48 / 48 * SWAP 12 * + >R
YCON @ 40 / 40 * OVER 20 * +
R> SWAP SETRELOIGIN
6 0 DO ( rows)
6 0 DO ( cols)
OVER I + OVER J + CONANCHOR LC@
I 12 * J 20 * CONTOUR LC!
LOOP LOOP 2DROP T;

```

11-19-84)

29

```

11-19-84) ( FRACT-OV ANCHOR-CONTOUR
: ANCHOR-CONTOUR ( -- \ place anchors )
( for sub-contour regions within the)
( contour array based on the mercator)
( map and xcon,ycon, the ul corner of)
( the contour map)
MERC>CONANCHOR
' CONANCHOR SETLARRAY
CONANCHOR-HOR
CONANCHOR-VER
CONANCHOR-CNT1
CONANCHOR-CNT2
CONANCHOR)CONTOUR ;

```

11-19-84)

30

```

0 ( FRACT-OV SUB-CON-FRACT
1 HEAD: SUB-CON-FRACT ( col row -- \ fractal)
2 ( sub-contour region col,row if null)
3 T: 20 * SWAP 12 * SWAP ( x y --)
4 OVER 1+ OVER 1+ ACELLADDR A@
5 FNULL =
6 IF OVER 12 + OVER 20 + SETREGION
7 FRACT-REGION
8 ELSE 2DROP THEN T;
9
10
11
12
13
14
15

```

31

```

0 ( FRACT-OV FRACT-CONTOUR
1 : FRACT-CONTOUR ( -- \ fractalize all)
2 ( sub-contour regions in the contour)
3 ( map that contain nulls. Anchors are)
4 ( assumed to be in place.)
5 SPHEREWRAFF OFF SIGNEXTEND ON
6 ' CONTOUR SETLARRAY
7 XYANCHOR ON
8 CONTOUR-RATIO CONTOUR-SCALE
9 SETSCALE
10 5 0 DO ( for each row)
11 5 0 DO ( for each col)
12 I J SUB-CON-FRACT
13 LOOP LOOP ;
14
15

```

32

```

0 ( FRACT-OV +FRAME -FRAME @FRAME !FRAME
1 V: FRAME ( stack frame pointer) HEX
2 V: +FRAME -2 DP +! ( set stack frame) HERE DUP 2- !
3 89 C, 26 C, FRAME , AD C, 8B C, DB C, FF C, 27 C,
4 V: -FRAME -2 DP +! ( drop stack frame) HERE DUP 2- !
5 83 C, C4 C, 0E C, 89 C, 26 C, FRAME , AD C, 8B C, DB C,
6 FF C, 27 C,
7 V: @FRAME -2 DP +!
8 ( b -- n \ get word at [FRAME]+b) HERE DUP 2- !
9 5B C, 03 C, 1E C, FRAME , FF C, 37 C, AD C, 8B C, DB C,
10 FF C, 27 C,
11 V: !FRAME -2 DP +!
12 ( n b -- \ store word at [FRAME]+b) HERE DUP 2- !
13 5B C, 03 C, 1E C, FRAME , 8F C, 07 C, AD C, 8B C, DB C,
14 FF C, 27 C,
15 DECIMAL

```

33

```

11-19-84) ( FRACT-OV @X1 @X2 @XMID @Y1 @Y2 @YMID !XMID !YMID... 8-13-85)

HEAD: @X1 T: 6 @FRAME T;
HEAD: @X2 T: 2 @FRAME T;
HEAD: @XMID T: 10 @FRAME T;
HEAD: @Y1 T: 4 @FRAME T;
HEAD: @Y2 T: 0 @FRAME T;
HEAD: @YMID T: 8 @FRAME T;
HEAD: @STD T: 12 @FRAME T;
HEAD: !XMID T: 10 !FRAME T;
HEAD: !YMID T: 8 !FRAME T;
HEAD: !STD T: 12 !FRAME T;

HEAD: MIDPTT ( -- \ compute midpoint for current frame)
T: @X1 @X2 + 2/ !XMID @Y1 @Y2 + 2/ !YMID T;

```

34

```

11-19-84) ( FRACT-OV - OUTPOLY
11-26-84)

V: POLYPTR V: POLYCOUNTER
V: ULY V: ULX

HEAD: OUTPOLY ( -- \ save current polygon in the display list)
T: POLYPTR @ >R @STD POLYSEG I LC! ( surface byte)
@X1 ULX @ - POLYSEG I 1+ LC!
ULY @ @Y1 - POLYSEG I 2+ LC! ( upper left cor)
@X2 ULX @ - POLYSEG I 3+ LC!
ULY @ @Y2 - POLYSEG R> 4+ LC! ( lower right cor)
5 POLYPTR +! 1 POLYCOUNTER +! T;

```

35

```

8-13-85) ( FRACT-OV - ?HOMOGENOUS rfg03jun85)

HEAD: ?HOMOGENOUS
( -- t \ is currnt region all same surface type?)
T: @X1 @Y1 1- MERCATOR L+-@ COLORMAP
COLOR @ !STD ( get & save color)
@Y1 @Y2 DO
@X2 @X1 DO I J MERCATOR L+-@ COLORMAP color @ @STD = NOT
IF 999 !STD LEAVE THEN ( end on 1st exception)
LOOP
@STD 999 = IF LEAVE THEN
LOOP
@STD 999 = NOT T;

```


36

39

```

0 ( FRACT-OV - POLYGON-EXTRACT          8-13-85 ) ( FRACT-OV - CACCUM          8-13-85)
1
2 : POLYGON-EXTRACT ( 0 0 0 X1 Y1 X2 Y2 -- )
3 +FRAME                                ( set stack frame)
4 ?HOMOGENOUS                          ( is polygon solid?)
5 IF OUTPOLY                          ( if so, output poly)
6 ELSE MIDPTT                          ( else, subdivide...)
7   0 0 0 @X1 @Y1 @XMID @Y2 MYSELF      ( &try again)
8   0 0 0 @XMID @Y1 @X2 @Y2 MYSELF
9   0 0 0 @X1 @Y2 @XMID @Y2 MYSELF
10  0 0 0 @XMID @Y2 @X2 @Y2 MYSELF
11 THEN
12 -FRAME ;                            ( clear stack frame)
13
14
15

```

37

40

```

0 ( FRACT-OV - BUILD-CONTINENTS          10-07-85 ) ( FRACT-OV - MAJOR-COLOR          8-13-85)
1
2 HEAD: BUILD-CONTINENTS
3   ( -- \ build polygon list for all faces)
4 T: POLYPTR OFF
5   72 0 DO 0 0 0
6     I 12 /MOD ( rem y --)
7     6 swap - 2* 2* DUP ULY ! SWAP
8     4 * DUP ULX ! ( y x -- upper left)
9     SWAP OVER 4 + OVER 4 - POLYCOUNTER OFF
10    POLYPTR @ 0 I FACE L!
11    POLYGON-EXTRACT POLYCOUNTER @ 2 I FACE LC!
12    LOOP T;
13
14
15

```

38

41

```

0 ( FRACT-OV - +TMP -TMP @TMP !TMP          8-13-85 ) ( FRACT-OV - DROP-POLY PUSH-POLY POP-POLY          8-13-85)
1 V: [TMP] ( temp variable base pointer) HEX
2 V: +TMP -2 DP +! ( #cells -- \ reserve temp wkspce in stk)
3 HERE DUP 2- ! 89 C, 26 C, [TMP], 58 C, D1 C, E0 C, 28 C,
4 E0 C, AD C, 8B C, D8 C, FF C, 27 C,
5 V: -TMP -2 DP +! ( #cells -- \ reclaim temp wkspce in stk)
6 HERE DUP 2- ! 58 C, D1 C, E0 C, 03 C, E0 C, AD C, 8B C, D8 C,
7 FF C, 27 C,
8 V: @TMP -2 DP +! ( cell# -- val \ @ val of temp cell#) HERE DUP
9 2- ! 58 C, D1 C, E3 C, F7 C, D8 C, 03 C, 1E C, [TMP], FF C,
10 37 C, AD C, 8B C, D8 C, FF C, 27 C,
11 V: !TMP -2 DP +! ( val cell# -- \ save val in cell#) HERE DUP
12 2- ! 58 C, D1 C, E3 C, F7 C, D8 C, 03 C, 1E C, [TMP], 58 C,
13 89 C, 07 C, AD C, 8B C, D8 C, FF C, 27 C,
14 DECIMAL
15

```


42

45

```

0 ( FRACT-OV - ?MCLR MAINPOLY          8-13-85) ( FRACT-OV - UPSCRL DNSCRL          12-17-84)
1 HEX
2 V: ?MCLR -2 DP +!
3      ( color-byte -- color-byte t \ main color?)
4  HERE DUP 2- ! 33 C, C0 C, 58 C, 50 C, 38 C, 06 C, MAINCOLOR , T: 0 0 ACELLADDR      ( n sseg soff --)
5  75 C, 05 C, 88 C, 01 C, 00 C, EB C, 02 C, 33 C, C0 C, 50 C,   ROT 0 SWAP ACELLADDR ( sseg soff dseg doff -- )
6  AD C, 88 C, D8 C, FF C, 27 C,   3 PICK OVER - ABS      ( sseg soff dseg doff #bytes-not-moved)
7  V: MAINPOLY -2 DP +!   #BYTES SWAP -      ( sseg soff dseg doff count -- )
8      ( -- [4 4 0 0 color] \ build main polygon)   <LCMOVE T;
9  HERE DUP 2- ! 33 C, C0 C, 50 C, 88 C, 04 C, 00 C, BA C, E0 C,
10 50 C, 88 C, 06 C, MAINCOLOR , 50 C, AD C, 88 C, D8 C, FF C, HEAD: DNSCRL ( n -- \ scroll current array up n rows)
11 27 C,   T: 0 SWAP ACELLADDR ( sseg soff -- )
12 DECIMAL   0 0 ACELLADDR      ( sseg soff dseg doff -- )
13   3 PICK OVER - ABS ( sseg soff dseg doff #bytes-not-moved)
14   #BYTES SWAP -      ( sseg soff dseg doff count -- )
15   LCMOVE T;

```

43

46

```

0 ( FRACT-OV - REDUCE-FACE          11-26-84 ) ( FRACT-OV - RTSCRL LFSCRL          12-17-84)
1
2 HEAD: REDUCE-FACE ( face# -- \ remove redundant polygons)   HEAD: RTSCRL ( n -- \ scroll right current array n cols)
3 T: 0 OVER FACE OVER OVER L@ TAD ! 2+ LC@ ( face# polygon-cnt- ) T: #ROWZ 0 DO      ( for each row)
4 >R TAD @ I MAJOR-COLOR MAINCOLOR !      ( face# -- count)   0 1 ACELLADDR      ( n sseg soff --)
5 I 0 DO TAD @ PUSH-POLY 5 TAD +! LOOP ( face# -- count)   3 PICK I ACELLADDR      ( n sseg soff dseg doff --)
6 FADDR @ TAD ! MAINPOLY 1 #POL ! POP-POLY   #COLZ 6 PICK - <LCMOVE
7 R> 0 DO ?MCLR IF DROP-POLY   LOOP DROP T;
8      ELSE POP-POLY 1 #POL +!
9      THEN
10     LOOP
11 0 SWAP FACE OVER OVER TAD @ ROT ROT L!      ( addr --)
12 2+ #POL @ ROT ROT LC! T;
13
14
15

```

44

47

```

0 ( FRACT-OV - REDUCE-ALL MAKE-GLOBE          11-26-84 ) ( FRACT-OV - SCRLXCON          12-17-84)
1
2 HEAD: REDUCE-ALL
3      ( -- \ reduce # of polygons & reclaim dictionary)
4 T: 0 FADDR !      ( set initial polygon list addr)
5 72 0 DO I REDUCE-FACE LOOP T;
6
7 : MAKE-GLOBE BUILD-CONTINENTS REDUCE-ALL ;
8
9
10
11
12
13
14
15

```

```

HEAD: SCRLXCON ( -- \ scroll the contour map left or right and)
( null the area that is to be regenerated.)
T: DXCON @
IF DXCON @ DUP 0<
IF ABS RTSCRL 0 0 11 100
ELSE LFSCRL 49 0 60 100
THEN
SETREGION FNULL FILLREGION
THEN T;

```

48

51

```

0 ( FRACT-OV - SCRLYCON                                12-17-84)
1
2 HEAD: SCRLYCON ( -- \ scroll the contour map up or down and)
3 ( null the area that is to be regenerated.)
4 T: DYCON @
5 IF DYCON @ DUP 0<
6   IF ABS UPSCRL 0 0 60 19
7   ELSE DNSCRL 0 81 60 100
8   THEN
9   SETREGION FNULL FILLREGION
10 THEN T;
11
12
13
14
15

```

49

52

```

0 ( FRACT-OV - SCRLCON                                12-17-84)
1
2 : SCRLCON ( -- \ scroll the contour map to put the viewing )
3 ( frame back on the contour map.)
4 DXCON @ DYCON @ OR
5 IF SPHEREWRAP OFF
6   ' CONTOUR SETLARRAY
7   SCRLXCON ( scroll and null region)
8   SCRLYCON ( scroll and null region)
9   ANCHOR-CONTOUR
10  FRACT-CONTOUR
11 THEN ;
12
13
14
15

```

50

53

```

0 ( FRACT-OV ----- OVERLAY SUFFIX -----rfq03jun85)
1 trace @ trace off dispose trace !
2 CLOSE-OVERLAY
3 71 OVERLAY FRACT-OV
4 FRACT-OV
5 FORTH DEFINITIONS
6 : MERCATOR-GEN ( seed -- ) FRACT-OV FRACT MERCATOR-GEN ;
7 : INIT-CONTOUR ( -- ) FRACT-OV FRACT INIT-CONTOUR ;
8 : ANCHOR-CONTOUR ( -- ) FRACT-OV FRACT ANCHOR-CONTOUR ;
9 : FRACT-CONTOUR ( -- ) FRACT-OV FRACT FRACT-CONTOUR ;
10 : MAKE-GLOBE ( -- ) FRACT-OV FRACT MAKE-GLOBE ;
11 : SCRLCON ( -- ) FRACT-OV FRACT SCRLCON ;
12
13 OV-CANCEL
14 \ ' MERCATOR-GEN ' cM-GEN ! \ patch forward ref from shpmov-ov
15 \ cr ." cM-GEN is set to " cM-gen u.

```