

0 \ OVERLAY PREFIX: SETTING SPECIES AFIELDS
 1 \ constants and utility words
 2
 3 \ start of phrase to iaddr retrieval
 4 \ localized search
 5 \ scanning word for finding a particular phrase
 6 \ actual s" word compiles double literal instance
 7 \ >PROP gets proportion phrase iaddr, given index
 8 \ size lookup array
 9 \ look up a color iaddr, given its order (index)
 10 \ >PROP gets proportion phrase iaddr, given index
 11 \ curve and initial bit shuffling
 12 exit
 13 \ PERCENTAGE ARRAYS FOR COMPUTING EXISTENCE OF LIF
 14 \ attribute fields for planets
 15 \ orders fields
 16 \ continue orders
 17 \ species fields
 18 \ continue species fields
 19 \ continue species fields
 20 \ finish species description
 21 \ whichguy force current CREATURE attribute file
 22 \ PROBABILITY ARRAY : CURRENTLY ONE OF A KIND
 23 \ store an offset into prob array header
 24 (CASE STATEMENT BY CHARLES EAKER)
 25 (RESULT FROM ACCESSING PROBABILITY ARRAY
 26 \ maybe and =chance for probability arrays
 27 \ keys data structure, similar to probs
 28 \ retrieve, given %,row or %,-1
 29 \ probs for structures added to decorations
 30 \ extensions
 31 \ continue with structure probabilities
 32 \ continue with structure probabilities
 33 \ continue with structure probabilities
 34 \ continue with structure probabilities
 35 \ continue with structure probabilities
 36 \ finish structure probabilities
 37 \ color selection for all plants orders
 38 \ squish for class proportion setting
 39 \ setting species aggression level, using order in
 40 \ species hitpoints
 41 \ setting ground movement mode
 42 \ select proportion modifiers for species' feature
 43 \ add tint and lumens, given a hue to try
 44 \ colorants for species surface
 45 \ setting icon for species display
 46 \ icons for bilateral beings
 47 \ icons for bilateral animals
 48 \ now set the icon into the species
 49 \ colorants for species surface
 50 \ size to height conversion table
 51 \ highness
 52 \ choosing a size for all features
 53 \ setting species intelligence level, using order
 54 \ add structure to features
 55 \ start of simili determination

55 \ start of simili determination
 56 \ continue with simili
 57 \ continue with some animal similit test
 58 \ ?something (else)
 59 \ finish ?something
 60 \ final similarities word
 61 \ creating species information
 62 \ OVERLAY SUFFIX: SETTING SPECIES AFIELDS

0

3

```

0 \ OVERLAY PREFIX: SETTING SPECIES AFIELDS
1 ov-cancel 9 width !
2 vocabulary loparms immediate
3
4 90 open-overlay
5 loparms definitions
6
7 2600 trans-allot
8 newt-dp
9
10
11
12
13
14
15

```

```

\ start of phrase to iaddr retrieval
transient trace @ trace off
V= found 1 allot

: scan" \ parse to " --- addr,count
  ascii " word count 2dup >uppercase ;

: ?match
0 dup found 1.5!
dup phr-cnt c@ =
  if CI found 1.5! phrase phr-cnt c@ 0 do
    3 pick i + c@ over i + c@ -
    if 0 DUP found 1.5! leave then loop drop
  then ;

```

1

4

```

0 \ constants and utility words
1 decimal
2
3 32 c: planet
4 68 c: creature
5 transient
6 : PRESENT BOX-IADDR 1.5@ >c+s IOPEN ;
7 resident
8 head: difind t: 2dup ifind t;
9
10 head: ?2dup t: 2dup or if 2dup then t;
11
12
13
14
15

```

```

\ localized search
\ still uses ?first exit test, so doesn't wrap
v= (Starting) 1 allot

: defaultStart
  file# @ record# @ >r >r
  present ci (starting) 1.5! cdrop iclose
  r> r> record# ! file# ! ;

defaultstart

: startAt
  box-iaddr 1.5@ >c (starting) 1.5@ >c+s ;

```

2

5

```

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

```

```

\ scanning word for finding a particular phrase

: (s") 0 dup found 1.5!
record# @ >r file# @ >r
' nop 'map ! ' inext 'travers ! ' ?>first 'exit !
startat scan"
begin ?match inext ?first found 1.5@ or or until
found 1.5@ - 0=
if ." Phrase " type ." not found" abort then
2drop cdrop iclose found 1.5@
r> file# ! r> record# ! ;

\ leaves instance address on top of stack; use >c+s!!

```

6

```

0 \ actual s" word compiles double literal instance addresses
1 \ follow s" with the phrase object to be found
2 : s" (s") [compile] 2literal ; immediate
3
4 : start" [compile] s" (starting) 1.5! ;
5 resident trace !
6
7 head: rput t: record# @ file# @ t;
8 head: rget t: file# ! record# ! t;
9
10 head: 1.5, t: here 3 allot 1.5! t;
11
12 head: iaddr-array \ run-time: ord --- addr ! contains iaddr
13 t: create 0 1.5,
14 does> swap 3# + t;
15

```

7

```

0 \ >PROP gets proportion phrase iaddr, given index
1 head: lookup \ iaddr,array --- index or error report
2 t: rput >r >r dup c@ 3* 3+ ( get size of array )
3 3 do dup i + 1.5@
4 5 pick 5 pick d=
5 if drop 2drop 0. i leave then
6 3 +loop \ leaves 0.,index or iaddr,array
7 >r ?2dup or
8 if >c+s phrase phr-cnt c@ type
9 iclose ." not in " r> nfa id.
10 unravel quit
11 then r> 3 / r> r> rget t; \ leaves the index, hopefully
12
13
14
15

```

8

```

0 \ size lookup array
1 start" gargantuan"
2 9 iaddr-array >size
3 s" tiny" 1.5, s" small" 1.5, s" big" 1.5,
4 s" large" 1.5, s" very large" 1.5, s" huge" 1.5,
5 s" massive" 1.5, s" humongous" 1.5, s" gargantuan" 1.5,
6
7 defaultstart start" leaping"
8 10 iaddr-array >mover
9 s" oozing" 1.5, s" creeping" 1.5, s" slithering" 1.5,
10 s" crawling" 1.5, s" rolling" 1.5, s" moving" 1.5,
11 s" walking" 1.5, s" hopping" 1.5, s" jumping" 1.5,
12 s" leaping" 1.5,
13 defaultstart
14
15

```

9

```

\ look up a color iaddr, given its order (index)

defaultstart start" black"
27 iaddr-array ?color
s" red" 1.5, s" maroon" 1.5, s" rust-colored" 1.5,
s" pink" 1.5, s" reddish orange" 1.5, s" brown" 1.5,
s" beige" 1.5, s" copper-colored" 1.5, s" orange" 1.5,
s" yellowish orange" 1.5, s" yellow" 1.5, s" lemon yellow" 1.5,
s" yellowish green" 1.5, s" brownish green" 1.5, s" green" 1.5,
s" lime green" 1.5, s" blue-green" 1.5, s" turquoise" 1.5,
s" blue" 1.5, s" lavender" 1.5, s" violet" 1.5,
s" purple" 1.5, s" white" 1.5, s" multi-colored" 1.5,
s" grey" 1.5, s" black" 1.5, s" very blue" 1.5,
defaultstart

```

10

```

\ >PROP gets proportion phrase iaddr, given index

defaultstart start" delicate"
13 iaddr-array >prop
s" flattened" 1.5, s" flat" 1.5,
s" wide" 1.5, s" thick" 1.5, s" squat" 1.5,
s" elongated" 1.5, s" narrow" 1.5, s" slender" 1.5,
s" graceful" 1.5, s" thin" 1.5, s" spindly" 1.5,
s" willowy" 1.5, s" delicate" 1.5,
defaultstart

```

11

```

\ curve and initial bit shuffling
decimal
head: COIN t: 0 2 rrnd t;
head: RAND% t: 1 100 rrnd t;
head: crandom t: 0 256 rrnd t;

head: scale t: >r 1+ over - r> 256 */ + t;
head: slipper t: crandom (slipper) t;
head: slip! t: begin slipper until scale t;

```

rfg14jun85)

12

0 exit
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

13

0 \ PERCENTAGE ARRAYS FOR COMPUTING EXISTENCE OF LIFE
1
2 head: %%
3 t: CREATE 0 ,
4 DOES> OVER + + @ t;
5
6
7 (THIS IS USED TO BUILD ARRAYS WHICH CONTAIN PERCENTAGES
8 THESE %S ARE USED TO ADD OR SUBTRACT FROM A BASE % FOR A
9 TEMPERATURE REGION. THE RESULTING PERCENTAGE, WHEN COMPARED
10 TO A RANDOMLY DERIVED % , WILL DETERMINE IF THERE IS LIFE
11 ON THE PLANET AS A WHOLE, AND ALSO IF THERE IS LIFE IN A
12 PARTICULAR TEMPERATURE REGION.)
13
14
15

14

0 \ attribute fields for planets
1
2 \ planet 4 1 afield: plan-life
3 planet 19 1 afield: atmo.density
4 planet 20 3 afield: narrowest
5 planet 21 2 afield: flattest
6
7
8
9
10
11
12
13
14
15

15

\ orders fields rfg14jun85)
\ creature 0 2 afield: cllo \ lo record# for this class
\ creature 2 2 afield: clhi \ hi record#
creature 4 3 afield: shape \ token
creature 7 3 afield: exterior \ token
creature 11 1 afield: size.index
creature 12 1 afield: niche
creature 13 3 afield: reddest
creature 16 3 afield: bluest
creature 19 3 afield: texture
creature 22 3 afield: w.append
creature 25 1 afield: w#
creature 26 3 afield: d.append
creature 29 1 afield: d#

16

\ continue orders
creature 30 3 afield: a.append
creature 33 1 afield: a#
creature 34 3 afield: g.append
creature 37 1 afield: g#
creature 38 3 afield: g.mobility
creature 41 3 afield: a.mobility
\ creature 44 2 afield: orlo \ lo record# for an order
\ creature 46 2 afield: orhi \ hi record#

17

\ species fields rfg14jun85)
creature 48 1 afield: slowest
creature 49 1 afield: fastest
creature 51 1 afield: stun.effect
creature 52 1 afield: hit.points
creature 54 1 afield: i.level \ intelligence
creature 55 1 afield: a.level \ aggression
creature 56 2 afield: height
creature 58 2 afield: volume
creature 60 2 afield: proportion

18

```

0 \ continue species fields
1 creature 63 3 afield: d.hue
2 creature 66 3 afield: d.tint
3 creature 69 3 afield: d.lumens
4 creature 72 3 afield: d.struc
5 creature 75 3 afield: d.propor
6 creature 78 3 afield: d.size
7
8 creature 81 3 afield: w.hue
9 creature 84 3 afield: w.tint
10 creature 87 3 afield: w.lumens
11 creature 90 3 afield: w.struc
12 creature 93 3 afield: w.propor
13 creature 96 3 afield: w.size
14
15

```

19

```

0 \ continue species fields
1 creature 99 3 afield: a.hue
2 creature 102 3 afield: a.tint
3 creature 105 3 afield: a.lumens
4 creature 108 3 afield: a.struc
5 creature 111 3 afield: a.propor
6 creature 114 3 afield: a.size
7
8 creature 117 3 afield: g.hue
9 creature 120 3 afield: g.tint
10 creature 123 3 afield: g.lumens
11 creature 126 3 afield: g.struc
12 creature 129 3 afield: g.propor
13 creature 132 3 afield: g.size
14
15

```

20

```

0 \ finish species description
1 creature 135 3 afield: s.hue \ surface hue
2 creature 138 3 afield: s.tint
3 creature 141 3 afield: s.lumens
4 creature 144 2 afield: icon
5 creature 146 3 afield: resembles
6 creature 149 3 afield: size
7
8
9
10
11
12
13
14
15

```

21

```

rfg14jun85) \ whichguy force current CREATURE attribute file
              transient trace @ trace off
              : 20F          4 ?PAIRS COMPILE 20VER COMPILE d= COMPILE
                      0BRANCH HERE 0 , COMPILE 2DROP 5 ; IMMEDIATE
              resident trace !

```

```

head: whichguy \ force current attribute file to be referent
t: creature dup file# ! record# @ @record drop t;

```

22

```

rfg14jun85) \ PROBABILITY ARRAY : CURRENTLY ONE OF A KIND
head: MAPS \ [CELL W/LENGTH OF MAP] --- [H1] [L0] FOR DO...LOOP
t: DUP @ 2+ RANGE 2+ t;

head: PROBS \ <N> PROBS , N IS NUMBER OF (SINGULAR) ROW VALUES
t: CREATE 6 * DUP , here over allot swap 0 fill
DOES> 0 4 roll 4 roll 4 roll MAPS
      DO I 2@ 3 pick = >r 3 pick = r> and
          IF I 4 roll DROP rot rot LEAVE THEN
      6 +LOOP 2DROP ?DUP
      IF 4 + @ ELSE
      ." INDEX NOT IN PROBABILITY ARRAY! " unravel quit
      THEN t;

```

23

```

rfg14jun85) \ store an offset into prob array header
              transient trace @ trace off
              decimal variable %total 100 %total !

              : %remain 100 %total @ - ;
              : mark \ row value ---
                  latest pfa maps
                  do i 2@ or 0= if i leave then 6 +loop
                  here over 4 + ! 2!
                  0 %total ! ;

              : %, \ result, 0<=probability<=100 ---
                  %total over over @ + c, +! \ compiled prob. is a byte
                  1.5, ; \ the stored result may be a literal, pfa, etc..
              : Op 0 %, ;

```

24

```

0 ( CASE STATEMENT BY CHARLES EAKER )
1 DECIMAL
2 : !CSP SP@ CSP ! ;
3 : ECASE      ?COMP CSP @ !CSP 4 ; IMMEDIATE
4
5 : EOF        4 ?PAIRS COMPILE OVER COMPILE = COMPILE
6              OBRANCH HERE 0 , COMPILE DROP 5 ; IMMEDIATE
7
8 : ENDOF      5 ?PAIRS COMPILE BRANCH HERE 0 ,
9              SWAP 2 [COMPILE] THEN 4 ; IMMEDIATE
10
11 : ENDCASE    4 ?PAIRS COMPILE DROP BEGIN SP@
12             CSP @ = 0= WHILE 2 [COMPILE]
13             THEN REPEAT CSP ! ; IMMEDIATE
14 resident trace !
15

```

25

```

0 ( RESULT FROM ACCESSING PROBABILITY ARRAY      01/10/84 ) \ retrieve, given %,row or %, -1
1 decimal
2
3 head: CHANCE \ [SAMPLE BYTE] [KEYED ADDRESS] --- [PROB CELL]
4 t: BEGIN OVER OVER C@
5 DUP 100 = NOT >R
6   > R) AND
7 WHILE 1+ 3+ \ SKIP PAST THIS PROB. AND RESULT
8 REPEAT t;
9
10 head: RESULT t: CHANCE 1+ 1.5@ rot DROP t;
11
12 \ THE RESULT MAY BE A VALUE, ADDRESSE, ETC.
13 \ for niches THE PROBABILITY ROW IS 0<=AN OFFSET<=15
14
15

```

26

```

0 \ maybe and =chance for probability arrays
1 transient trace @ trace off
2 : maybe \ row# --- start address in multirow prob array
3   mark here 2 ;
4 variable fraction variable quotient
5
6 : =chance \ start addr
7   2 ?pairs 1 %total ! \ initial counter
8 here swap over over - 300 over /mod
9 quotient ! 100 * swap / dup fraction ! rot rot
10 do quotient @ over 49 > if 1+ then %total +!
11   %total @ 100 min i c! fraction @ + 100 mod \ patch Op's
12 4 +loop drop
13 100 here 4 - c! ; \ fudge final
14
15

```

27

```

\ keys data strcuture, similar to probs      rfg18jul84

: keys \ <N> keys , N IS NUMBER OF (SINGULAR) ROW VALUES
CREATE 6 * DUP , here over allot swap 0 fill ;

resident trace !

head: rollem \ %,key --- -1 or %,address
t: 0 4 roll 4 roll 4 roll
  maps DO I 2@ 3 pick = >r 3 pick = r) and \ find key in a
    IF I 4 roll DROP rot rot leave THEN \ lose 0 flag
    6 +LOOP 2DROP ?DUP \ lose the index key
    IF 4 + @ \ get address of row with %,result cells
    ELSE -1 \ key not found \ ignore in case of ADDITIONS
    THEN t;

```

28

```

head: retrieve
t: rollem dup -1 =
  if 2drop 0 0 \ return 'undefined' as field value
  else result \ or get a modifier from this probability row
  then t;

```

29

```

\ probs for structures added to decorations      rfg09jul84
26 keys additions start" head"
s" leg" maybe s" stalk-like" Op s" long" Op s" multi-jointed" Op
s" jointed" Op s" muscular" Op s" powerful looking" Op
s" stubby" Op s" short" Op =chance
s" tentacle" maybe s" gelatinous" Op s" whip-like" Op
s" sinewy" Op s" gnarled" Op s" spiny" Op s" membranous" Op
s" branched" Op s" muscular" Op
s" stubby" Op =chance
s" wing" maybe s" bat" Op s" butterfly" Op s" vestigial" Op
s" fin-like" Op s" floppy" Op s" fan-shaped" Op s" webbed" Op
s" sweeping" Op s" curved" Op s" leaf-like" Op s" feathery" Op
s" rigid" Op s" irregularly-shaped" Op s" membranous" Op
s" jointed" Op s" muscular" Op s" intricate" Op
s" powerful looking" Op s" stubby" Op =chance

```

30

```

0 \ extensions
1 s" float sac" maybe s" sphere-shaped" Op s" globular" Op
2 s" dome-shaped" Op s" tear-shaped" Op s" bulbous" Op s" rigid"
3 Op s" irregularly-shaped" Op s" spiny" Op s" membranous" Op
4 s" feathery" Op s" bulging" Op s" leaf-like" Op =chance
5 s" arm" maybe s" gnarled" Op s" spiny" Op s" multi-jointed" Op
6 s" jointed" Op s" branched" Op s" muscular" Op
7 s" powerful looking" Op s" stubby" Op =chance
8 s" branches" maybe s" flexible" Op s" spear-shaped" Op
9 s" pronged" Op s" pointed" Op s" sweeping" Op s" curved" Op
10 s" rigid" Op s" irregularly-shaped" Op s" gnarled" Op
11 s" stubby" Op s" rod-shaped" Op s" spiny" Op =chance
12 s" horn" maybe s" spiral-shaped" Op s" spiked" Op
13 s" blunt" Op s" conical" Op s" hooked" Op s" razor sharp" Op
14 s" sharp" Op s" pronged" Op s" pointed" Op s" intricate" Op
15 s" needle-shaped" Op

```

31

```

0 \ continue with structure probabilities rfg09jul84
1 s" sweeping" Op s" curved" Op s" irregularly-shaped" Op
2 s" gnarled" Op s" stubby" Op =chance
3 s" beak" maybe s" blunt" Op s" conical" Op s" hooked" Op
4 s" razor sharp" Op s" sharp" Op s" pointed" Op s" curved" Op
5 s" gnarled" Op s" powerful looking" Op s" stubby" Op =chance
6 s" claws" maybe s" hooked" Op s" razor sharp" Op s" sharp" Op
7 s" pointed" Op s" webbed" Op s" jointed" Op s" muscular" Op
8 s" powerful looking" Op =chance
9 s" teeth" maybe s" saber" Op s" fang-like" Op
10 s" blade-like" Op s" needle-shaped" Op s" razor sharp" Op
11 s" sharp" Op s" pointed" Op s" thorn-like" Op =chance
12 s" tail" maybe s" spiked" Op s" pointed" Op s" fan-shaped" Op
13 s" sweeping" Op s" curved" Op s" feathery" Op s" whip-like" Op
14 s" prehensile" Op s" spiny" Op s" muscular" Op
15 s" powerful looking" Op =chance

```

32

```

0 \ continue with structure probabilities rfg09jul84
1
2 s" feet" maybe s" splayed" Op s" cupped" Op s" dish-shaped" Op
3 s" disk-shaped" Op s" saucer-shaped" Op s" pointed" Op
4 s" floppy" Op s" fan-shaped" Op s" webbed" Op s" leaf-like" Op
5 s" feathery" Op s" prehensile" Op s" gnarled" Op
6 s" stubby" Op =chance
7 s" tusk" maybe s" protruding" Op s" fang-like" Op
8 s" hooked" Op s" pronged" Op s" pointed" Op
9 s" sweeping" Op s" curved" Op s" gnarled" Op
10 s" spiral-shaped" Op s" pointed" Op =chance
11 s" snout" maybe s" protruding" Op s" blunt" Op s" conical" Op
12 s" pointed" Op s" bulbous" Op s" prehensile" Op
13 s" membranous" Op s" muscular" Op s" stubby" Op =chance
14
15

```

33

```

\ continue with structure probabilities rfg09jul84
s" feeler" maybe s" protruding" Op s" blunt" Op s" conical" Op
s" pointed" Op s" bulbous" Op s" prehensile" Op
s" membranous" Op s" stubby" Op =chance
s" shell" maybe s" scalloped" Op s" cuboid" Op
s" round" Op s" octagonal" Op s" oval" Op
s" bowl-shaped" Op s" dish-shaped" Op
s" disk-shaped" Op s" saucer-shaped" Op s" gnarled" Op
s" spiral-shaped" Op s" conical" Op s" dome-shaped" Op
s" intricate" Op s" irregularly-shaped" Op s" spiked" Op
s" tear-shaped" Op s" spiny" Op s" jointed" Op =chance
s" protrusion" maybe s" projection" maybe
s" globular" Op s" bulging" Op s" bulbous" Op s" fin-like" Op
s" leaf-like" Op s" feathery" Op s" gelatinous" Op
s" spiny" Op s" membranous" Op s" stalk-like" Op
s" gnarled" Op s" branched" Op s" stubby" Op =chance

```

34

```

\ continue with structure probabilities rfg09jul84
s" crest" maybe s" finely sculpted" Op s" armoured" Op
s" spiked" Op s" fin-like" Op s" leaf-like" Op
s" spiny" Op s" membranous" Op s" feathery" Op =chance
s" ear" maybe s" hole-like" Op s" bowl-shaped" Op
s" cupped" Op s" pointed" Op s" floppy" Op s" fan-shaped" Op
s" feathery" Op s" membranous" Op s" protruding" Op =chance
s" eye" maybe s" deep set" Op s" pupil-less" Op s" hole-like" Op
s" oval" Op s" protruding" Op s" saucer-shaped" Op
s" bulging" Op =chance
s" flowers" maybe s" heart-shaped" Op s" hexagonal" Op
s" octagonal" Op s" bowl-shaped" Op s" cupped" Op
s" disk-shaped" Op s" retractable" Op s" saucer-shaped" Op
s" sphere-shaped" Op s" tear-shaped" Op s" fan-shaped" Op
s" finely sculpted" Op s" membranous" Op s" dish-shaped" Op
s" feathery" Op s" spiral-shaped" Op =chance

```

35

```

\ continue with structure probabilities rfg09jul84
s" fruits" maybe s" cuboid" Op s" round" Op s" oval" Op
s" disk-shaped" Op s" saucer-shaped" Op s" sphere-shaped" Op
s" globular" Op s" tear-shaped" Op s" bulbous" Op s" curved" Op
s" rod-shaped" Op s" gelatinous" Op s" gnarled" Op s" spiny" Op
=chance s" leaves" maybe s" diamond-shaped" Op s" square" Op
s" hexagonal" Op s" circular" Op s" triangular" Op
s" scalloped" Op s" round" Op s" octagonal" Op s" oval" Op
s" bowl-shaped" Op s" cupped" Op s" dish-shaped" Op
s" spiral-shaped" Op s" razor sharp" Op s" spear-shaped" Op
s" rod-shaped" Op s" pointed" Op s" globular" Op
s" floppy" Op s" intricate" Op s" curved" Op
s" irregularly-shaped" Op s" spiny" Op s" membranous" Op
s" feathery" Op s" heart-shaped" Op s" saucer-shaped" Op
s" disk-shaped" Op s" needle-shaped" Op s" tear-shaped" Op
s" fan-shaped" Op s" rigid" Op =chance

```


36

39

```

0 \ finish structure probabilities          rfg09jul84 \ setting species aggression level, using order info
1
2 s" head" maybe s" triangular" Op s" oval" Op s" retractible" Op head: madness \ niche --- lo hi
3 s" conical" Op s" globular" Op s" dome-shaped" Op s" bulbous" Op t: ecase ( producer) 02 eof 1 10 endof
4 s" gnarled" Op =chance                    ( herbivore) 04 eof 1 60 endof
5                                           ( herbivorous producer) 06 eof 1 60 endof
6 s" mouth" maybe s" cavernous" Op s" gaping" Op                    50 100 rot endcase t;
7 s" slash-like" Op s" circular" Op s" triangular" Op =chance
8 defaultstart
9
10 2 probs lights start" shimmering"
11 1. maybe s" dark" Op s" bright" Op s" light" Op =chance
12 2. maybe s" blinding" Op s" bright" Op s" glowing" Op
13 s" luminescent" Op s" translucent" Op s" shimmering" Op
14 =chance defaultstart
15

```

37

40

```

0 \ color selection forall plants orders    rfg20may8 \ species hitpoints          rfg08apr85
1
2
3 head: ?pure \ color object --- object,tflag if pure color
4 t: rput >r >r 2dup >c+s \ color object --- object,fla
5 @inst-species 7 = iclose >r> rget t;
6
7
8
9
10
11
12
13
14
15

```

38

41

```

0 \ squish for class proportion setting    rfg30apr85 \ setting ground movement mode    rfg09jun
1
2
3 head: squish
4 t: record# @ >r
5 (planet) 1.5@ >c+s
6 flattest c@ narrowest c@ 1+ rrnd iclose \ range of propors
7 dup \ save volume, gets replaced by real value in HIGHNESS
8 >prop 1.5@ \ get proportion with this index
9 r> record# ! whichguy proportion 1.5! volume ! t;
10
11
12
13
14
15

```


42

```

0 \ select proportion modifiers for species' features
1 head: (modify) \ ?propor,?append ---
2 t: ifld@ >r >r
3 0 dup i i' d=
4 r>r> s" tail" d= or not \ these appends can have mod
5 if ( flattened) 1 ( delicate+1 ) 14 rrnd
6 >prop 1.5@ rot ifld!
7 else drop
8 then t;
9
10 head: modify
11 t: ' d.propor ' d.append (modify) ' w.propor ' w.append (modify)
12 ' a.propor ' a.append (modify)
13 ' g.propor ' g.append (modify) t;
14
15

```

45

```

\ setting icon.for species display rfg09jun84

head: image1 \ assumes all markers are set (for radial/spherical
t: g# c@ ?dup 0=
if 12
else 3 - 0< if 13 else 14 then
then t;

head: image2 \ for amorphous/irregular
t: g# c@
if 16 else 15 then t;

```

43

```

0 \ add tint and lumens, given a hue to try rfg14sep84 \ icon.s for bilateral beings rfg24jun85)
1 \ below sets at species level; fields are 'ed and IFLD@ed 3 probs plant.icons
2 head: customize \ 'some luminosity, 'some tint, some hue@ --- s" sessile" maybe 0. maybe
3 t: ?pure >r 2drop r> if coin \ 0. 0p 1. 0p 2. 0p 3. 0p
4 if rand% 1 0 lights result rot ifld! drop 4. 0p 5. 0p =chance
5 else coin \ ditto above note s" crawling" maybe 4. 0p 5. 0p =chance
6 if drop rand% 2 0 lights result rot ifld!
7 else 2drop
8 then
9 then
10 else 2drop
11 then t;
12
13
14
15

```

46

```

head: image3 \ for bilateral plants
t: coin
g.mobility 1.5@ s" crawling" d=
if if 4 else 5 then
else 9 size.index c@ - 3 / 2 min 2* ( 0,2 or 4)
+ \ add coin flag
then t;

```

44

```

0 \ colorants for species surface rfg29m
1
2 head: colorant
3 t: record# @ >r reddest c@ bluest c@ 1+ rrnd ?color 1.5@
4 2dup s.hue 1.5!
5 0. s.tint 1.5! 0. s.lumens 1.5!
6 ' s.lumens ' s.tint r> record# ! whichguy 2swap customize t;
7
8
9
10
11
12
13
14
15

```

47

```

\ icons for bilateral animals rfg10jun84

head: image4 \ for bilateral animals
t: g# c@ ?dup 0=
if proportion 1.5@ s" elongated" d=
if 6 else 7 then
else ecase 1 eof 8 endof
2 eof 9 endof
3 eof 10 endof
4 eof 10 endof
11 swap endcase
then t;

```

48

51

0 \ now set the icon into the species

rfg09jun84 \ highness

rfg30apr85

1 head: set.icon

2 t: shape 1.5@ 2dup s" radial" d= >r

3 2dup s" spherical" d= r> or

4 if 2drop image1

5 else 2dup s" amorphous" d= >r s" irregular" d= r> or

6 if image2

7 else niche c@ 4 mod

8 if image3

9 else image4

10 then

11 then

12 then icon ! t;

13

14

15

head: highness

t: size.index c@ dup

shortest

swap niche c@ 4 mod

if stalker else tallest then 1+ rrnd

dup height !

volume @ \ hold proportion index

/ 1 max volume ! t;

\ thin creatures have their volume reduced

49

52

0 \ colorants for species surface

rfg29may84 \ choosing a size for all features

1

2 head: colorize \ 'lumens' tints 'hue' append --- set species

3 t: ifld@ or

4 if reddest c@ bluest c@ 1+ rrnd

5 ?color 1.5@ 2dup 5 roll ifld!

6 customize

7 else 0 dup rot ifld! 0 dup rot ifld! 0 dup rot ifld!

8 then t;

9

10 head: paint

11 t: 'a.lumens' 'a.tint' 'a.hue' 'a.append colorize

12 'g.lumens' 'g.tint' 'g.hue' 'g.append colorize

13 'd.lumens' 'd.tint' 'd.hue' 'd.append colorize

14 'w.lumens' 'w.tint' 'w.hue' 'w.append colorize t;

15

head: (size) \ appendage ---

t: ifld@ or if 1 size.index c@ 1+ rrnd >size 1.5@

else 0.

then rot ifld! t;

head: size

t: 'g.size' 'g.append (size)

'a.size' 'a.append (size)

'd.size' 'd.append (size)

'w.size' 'w.append (size) t;

50

53

0 \ size to height conversion table

1 %% tallest

2 10 , 35 , 80 , 180 , 300 , 600 , 1033 , 1366 ,

3 1733 , 2133 ,

4 %% shortest

5 1 , 08 , 25 , 55 , 106 , 133 , 160 , 186 , 213 ,

6 240 ,

7 %% stalker \ plant max height-still use shortest for min

8 20 , 70 , 220 , 700 , 1800 , 3700 , 5866 ,

9 8733 , 12433 , 17066 , \ want to keep as unsigned

10

11

12

13

14

15

\ setting species intelligence level, using order info

\ used madness case since the numbers are so similar

head: smarts

t: niche c@ madness 128 peak !

slip! i.level c! t;

head: set-size \ just copies size index (iaddr) into size

t: size.index c@ 1 max 9 min >size 1.5@ size 1.5! t;

54

```

0 \ add structure to features
1
2 head: sculpta
3 t: rand% d.append 1.5@ additions retrieve d.struc 1.5!
4 rand% w.append 1.5@ additions retrieve w.struc 1.5!
5 rand% g.append 1.5@ additions retrieve g.struc 1.5!
6 rand% a.append 1.5@ additions retrieve a.struc 1.5! t;
7
8 exit
9
10
11
12
13
14
15

```

55

```

0 \ start of simili determination
1 head: ?plant
2 t: g.append 1.5@ s" leaves" d=
3 if size.index c@ ( very large) 5 <
4 if s" plant-like" else s" tree-like" then
5 else a.append 1.5@ s" float sac" d= >r
6 a# c@ 1 = r> and
7 if s" balloon-like"
8 else g.append 1.5@ s" tentacle" d= >r
9 g# c@ 5 > r> and
10 if s" anemone-like"
11 else 2@ rand% <
12 if s" fungoid" else s" plant-like" then
13 then then then t;
14
15

```

56

```

0 \ continue with simili
1 head: ?icky
2 t: shape 1.5@ s" irregular" d=
3 exterior 1.5@ >r>r
4 i' i s" hard" d= i' i s" rough" d= i' i s" coarse" d=
5 i' i s" crusty" d= r>r> s" cracked" d= or or or or and
6 if s" rock-like"
7 else shape 1.5@ s" elongated" d=
8 g# c@ 0= and
9 if s" starfish-like"
10 else shape 1.5@ s" exoskeleton" d=
11 g# c@ dup 3 > swap 7 < and and
12 if s" insectoid"
13 else proportion 1.5@ s" narrow" d=
14 g.append 1.5@ s" leg" d= >r
15

```

57

```

rfg11jul84 \ continue with some animal similit test
g# c@ 4 > r> and and
if s" spider-like"
else g.append 1.5@ s" leg" d= >r
d.append 1.5@ s" arm" d= r> and
shape 1.5@ s" bilateral" d= and
if s" humanoid"
else 0
then then then then then t;

```

58

```

rfg18jul84 \ ?something (else)
rfg18jul84
head: ?something
t: exterior 1.5@ s" scales" d=
if size.index c@ ( big) 3 >
if s" dinosaur-like"
else d.append 1.5@ s" shell" d=
if s" turtle-like" else s" reptilian" then
then
else exterior 1.5@ s" feathery" d=
a.append 1.5@ s" wing" d= and
if s" bird-like"
else texture 1.5@ s" mucousy" d=
if d.append 1.5@ s" shell" d=
if s" snail-like" else s" slug-like" then

```

59

```

rfg18jul84 \ finish ?something
else exterior 1.5@ s" fur" d=
if size.index ( big) 3 <
if s" rodent-like"
else g.append 1.5@ s" leg" d= >r
g# c@ dup 3 = swap 4 = or r> and
if proportion 1.5@ 2dup 2dup
s" thick" d= 2swap s" squat" d= or
if drop s" bear-like"
else 0 >prop lookup ( slender) 8 >
if s" antelope-like"
else 0. then then
else 0. then then
else 0. then then then then t;

```


60

```
0 \ final similarities word          rfg18jul8
1
2 head: similarity
3 t:  niche c@ 4 mod
4   if ?plant
5   else ?icky ?dup 0=
6   if ?something then
7   then resembles 1.5! t;
8
9
10
11
12
13
14
15
```

61

```
0 \ creating species information
1
2
3 : spec.parms
4 whichguy record# @ seed !
5 set-size squish anger durable stunable movers set.icon
6 highness smarts sculpta similarity modify colorant
7 paint sizer ;
8
9
10
11
12
13
14
15
```

62

```
0 \ OVERLAY SUFFIX: SETTING SPECIES AFIELDS      rfg14jun85)
1 trace @ trace off dispose trace !
2 close-overlay
3 90 overlay lp-ov
4 lp-ov
5
6 forth definitions
7
8 \ module definitions
9 : other-parms \ sets species creatures afields
10 lp-ov loparms spec.parms save-buffers ;
11 save-buffers ov-cancel
12
13
14
15
```