

```

0 \ OVERLAY PREFIX: SETTING CLASS/ORDER AFIELDS
1 \ constants
2 \ phrased getter/debugger
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
8 \ find a phrase,species,ord object
9 \ find a phrase,species,ord object
10 \ test species of a phrase for singular or plural
11 ( RANDOM NUMBER GENERATOR FROM STARTING FORTH
12
13
14 \ PERCENTAGE ARRAYS FOR COMPUTING EXISTENCE OF LIF
15 \ attribute fields for planets
16 \ orders fields
17 \ continue orders
18 \ whichguy force current CREATURE attribute file
19 \ PROBABILITY ARRAY : CURRENTLY ONE OF A KIND
20 \ store an offset into prob array header
21 ( CASE STATEMENT BY CHARLES EAKER )
22 ( RESULT FROM ACCESSING PROBABILITY ARRAY
23 \ maybe and =chance for probability arrays
24 \ start of context object
25 \ "shape table"
26 \ probs for animal class surfacing
27 \ classes shaping routine
28 \ continue classes epiderm
29 \ generate CLASS.PARMS
30 \ probs arrays for animal orders ground appendages
31 \ textures for surfaces
32 \ continue surface textures
33 \ continue surface textures
34 \ continue (patch) surface textures
35 \ finish surface textures
36 \ weapon fetature probability array
37 \ decorative feature probability array
38 \ #of... legs
39 \ movement mode for beings with legs
40 \ orders sizing and setting class fields
41 \ color selection forall plants orders
42 \ color selection forall plants orders
43 \ select weapon
44 \ animate #of legs
45 \ flying generation
46 \ continue setting flying orders
47 \ plant mobility
48 \ continue with plants/animals
49 \ choose animals orders ground appendage (and plan
50 \ choosing ground movement mode
51 \ setting ground movement mode
52 \ animate #of legs
53 \ finish surface textures
54 \ copying order information: stack, retrieve and d
55 \ copying minimal information: regions and niches

```

```

55 \ copying minimal information: regions and niches
56 \ OVERLAY SUFFIX: SETTING CLASS/ORDER AFIELDS
57 \ BIO module caller

```

0

3

```

0 \ OVERLAY PREFIX: SETTING CLASS/ORDER AFIELDS   rfg14jun85) \ start of phrase to iaddr retrieval
1 vocabulary hiparms immediate                   transient
2                                                  2V= found
3 89 open-overlay
4 hiparms definitions                             : scan" \ parse to " --- addr,count
5                                                  ascii " word count 2dup >uppercase ;
6 2250 trans-allot
7 newt-dp                                         : ?match
8                                                  0 dup found 1.5!
9 dup phr-cnt c@ =
10 if CI found 1.5! phrase phr-cnt c@ 0 do
11   3 pick i + c@ over i + c@ -
12   if 0 DUP found 1.5! leave then loop drop
13   then ;
14
15
```

1

4

```

0 \ constants                                     rfg14jun85) \ localized search
1 decimal                                         \ still uses ?first exit test, so doesn't wrap
2                                                  2v= (Starting)
3 32 c: planet
4 24 c: star
5 67 c: regions
6 68 c: creature
7 regions iheadlen 2 ifield: crowding
8
9
10
11
12
13
14
15
```

```

: 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 \ phrased getter/debugger
1
2 head: difind t: 2dup ifind t;
3
4 transient
5 : PRESENT
6 BOX-IADDR 1.5@ >c+s IOPEN ;
7 resident
8
9
10
11
12
13
14
15
```

```

\ scanning word for finding a particular phrase
\ follow s" with the phrase object to be found
: (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
2 : s" (s") [compile] 2literal ; immediate
3 : start" [compile] s" (starting) 1.5! ;
4 resident
5
6
7
8
9
10
11
12
13
14
15

```

7

```

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

```

8

```

0 \ find a phrase,species,ord object
1 head: rput t: record# @ file# @ t;
2 head: rget t: file# ! record# ! t;
3
4 head: 1.5, t: here 3 allot 1.5! t;
5
6 head: iaddr-array \ run-time: ord --- addr ! contains iaddr
7 t: create
8 does> swap 3* + t;
9
10 exit
11
12
13
14
15

```

9

```

\ find a phrase,species,ord object

defaultstart
27 iaddr-array ?color
s" red" 1.5, 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" grey" 1.5, s" black" 1.5,
s" very blue" 1.5,

```

10

```

\ test species of a phrase for singular or plural

head: ?s \ phrase object# --- object#,tflag if plural only
t: rput >r >r \ save the creature file,record
  2dup >c+s phr-cnt c@ 1-
  phrase + c@ ascii S = \ check last letter of ph
  @inst-species 4 = or \ plural only ?
  iclose r> r> rget t; \ restore creature

head: ?. t: rput >r >r 2dup >c+s \ test for 'no dtrm
  @inst-species 5 = iclose r> r> rget t;

head: .? t: rput >r >r 2dup >c+s \ test for singular
  @inst-species 3 = iclose r> r> rget t;
\ above two, same parm. format as ?s

```

11

```

( RANDOM NUMBER GENERATOR FROM STARTING FORTH 12/23/83 )
DECIMAL

head: crandom t: 0 256 rrnd t;
head: COIN t: 0 2 rrnd t;
head: RAND% t: 1 100 rrnd t;
head: constrain \ lo,hi,value --- value clipped within lo,hi
t: min max t;

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

```

12

15

```

0      \ attribute fields for planets      rfg14jun85)
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      planet 2 2 afield: plan-mass
7
8
9
10
11
12
13
14
15

```

13

16

```

0      \ orders fields      rfg14jun85)
1      creature 0 2 afield: cllo \ lo record# for this class
2      creature 2 2 afield: clhi \ hi record#
3      creature 4 3 afield: shape \ token
4      creature 7 3 afield: exterior \ token
5      \ creature 10 1 afield region \ record# in REGIONS file
6      creature 11 1 afield: size.index
7      creature 12 1 afield: niche
8      creature 13 3 afield: reddest
9      creature 16 3 afield: bluest
10     creature 19 3 afield: texture
11     creature 22 3 afield: w.append
12     creature 25 1 afield: w#
13     creature 26 3 afield: d.append
14     creature 29 1 afield: d#
15

```

14

17

```

0  \ PERCENTAGE ARRAYS FOR COMPUTING EXISTENCE OF LIFE rfg14jun85)
1  v: probable
2  head: %likely t: probable ! t;
3  head: probably t: rand% probable @ < t;
4
5  head: %% t: CREATE 0 ,
6      DOES> OVER + + @ t;
7
8
9  ( THIS IS USED TO BUILD ARRAYS WHICH CONTAIN PERCENTAGES
10 THESE %S ARE USED TO ADD OR SUBTRACT FROM A BASE % FOR A
11 TEMPERATURE REGION. THE RESULTING PERCENTAGE, WHEN COMPARED
12 TO A RANDOMLY DERIVED % , WILL DETERMINE IF THERE IS LIFE
13 ON THE PLANET AS A WHOLE, AND ALSO IF THERE IS LIFE IN A
14 PARTICULAR TEMPERATURE REGION.)
15

```

```

\ continue orders      rfg14jun85)
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#

```

18

```

0 \ whichguy force current CREATURE attribute file
1 transient
2 : 20F          4 ?PAIRS COMPILE 2OVER COMPILE d= COMPILE
3             0BRANCH HERE 0 , COMPILE 2DROP 5 ; IMMEDIATE
4 resident
5 head: 2pick t: dup >r pick r> pick t;
6
7 head: ?2dup t: 2dup or if 2dup then t;
8
9 head: whichguy \ force current attribute file to be referent
10 t: creature dup file# ! record# @ @record drop t;
11
12
13
14
15

```

19

```

0 \ PROBABILITY ARRAY : CURRENTLY ONE OF A KIND
1 head: MAPS \ [CELL W/LENGTH OF MAP] --- [HI] [LO] FOR DO...LOOP
2 t: DUP @ 2+ RANGE 2+ t;
3
4 head: PROBS \ <N> PROBS , N IS NUMBER OF (SINGULAR) ROW VALUES
5 t: CREATE 6 * DUP , here over allot swap 0 fill
6 DOES> 0 4 roll 4 roll 4 roll MAPS
7 DO I 2@ 3 pick = >r 3 pick = r> and
8 IF I 4 roll DROP rot rot LEAVE THEN
9 6 +LOOP 2DROP ?DUP
10 IF 4 + @ ELSE
11 ." INDEX NOT IN PROBABILITY ARRAY! " unravel quit
12 THEN t;
13
14
15

```

20

```

0 \ store an offset into prob array header
1 transient
2 decimal variable %total 100 %total !
3 : %remain 100 %total @ - ;
4 : mark \ row value ---
5 latest pfa maps
6 do i 2@ or 0= if i leave then 6 +loop
7 here over 4 + ! 2!
8 0 %total ! ;
9
10 : %, \ result, 0<=probability<=100 ---
11 %total over over @ + c, +! \ compiled prob. is a byte
12 1.5, ; \ the stored result may be a literal, pfa, etc..
13 : 0p 0 %, ;
14
15

```

21

```

( CASE STATEMENT BY CHARLES EAKER )
DECIMAL
: !CSP SP@ CSP ! ;
: ECASE      ?COMP CSP @ !CSP 4 ; IMMEDIATE
: EOF        4 ?PAIRS COMPILE OVER COMPILE = COMPILE
             0BRANCH HERE 0 , COMPILE DROP 5 ; IMMEDIATE
: ENDOF      5 ?PAIRS COMPILE BRANCH HERE 0 ,
             SWAP 2 [COMPILE] THEN 4 ; IMMEDIATE
: ENDCASE    4 ?PAIRS COMPILE DROP BEGIN SP@
             CSP @ = 0= WHILE 2 [COMPILE]
             THEN REPEAT CSP ! ; IMMEDIATE

resident

```

22

```

( RESULT FROM ACCESSING PROBABILITY ARRAY      01/10/84 )
decimal
head: CHANCE \ [SAMPLE BYTE] [KEYED ADDRESS] --- [PROB CELL]
t: BEGIN OVER OVER C@
   DUP 100 = NOT >R
   > R> AND
   WHILE 1+ 3+ \ SKIP PAST THIS PROB. AND RESULT
   REPEAT t;

head: RESULT t: CHANCE 1+ 1.5@ rot DROP t;

\ THE RESULT MAY BE A VALUE, ADDRESSE, ETC.
\ for niches THE PROBABILITY ROW IS 0<=AN OFFSET<=15

```

23

```

\ maybe and =chance for probability arrays
transient
: maybe \ row# --- start address in multirow prob array
   mark here 2 ;
variable fraction variable quotient

: =chance \ start addr
   2 ?pairs 1 %total ! \ initial counter
   here swap over over - 300 over /mod
   quotient ! 100 * swap / dup fraction ! rot rot
   do quotient @ over 49 > if 1+ then %total +!
   %total @ 100 min i c! fraction @ + 100 mod \ patch 0p's
   4 +loop drop
   100 here 4 - c! ; \ fudge final

resident

```



24

27

```

0 \ start of context object          rfg13apr84 \ classes shaping routine
1 decimal
2
3                                     head: shaper
4 head: (kng) t: 4 mod if plants else animals then t;    t: rand% niche c@ (kng) 0 shapes result
5                                     shape 1.5! t;
6
7
8
9
10
11
12
13
14
15

```

25

28

```

0 \ "shape table"                    rfg18may84 \ continue classes epiderm          rfg19may84
1 start" bilateral"
2 2 probs shapes animals 0 mark      head: outfit \ --- [exterior]
3 s" amorphous" 10 %, s" irregular" 10 %, s" spherical" 15 %, t: shape 1.5@
4 s" radial" 25 %, s" bilateral" 40 %,    begin 2dup rand% rot rot epiderm result
5                                     over 1+ ?dup 0= if drop 1+ then
6 plants 0 mark                        until >r >r 2drop r> r> t;
7 s" amorphous" 5 %, s" irregular" 5 %, s" spherical" 5 %,
8 s" radial" 30 %, s" bilateral" 55 %,
9 defaultstart                        head: pelts \ runs at classes level
10                                     t: niche c@ (kng) animals =
11                                     if outfit
12                                     else s" surface"
13                                     then exterior 1.5! t;
14
15

```

26

29

```

0 \ probs for animal class surfacing  rfg18may \ generate CLASS.PARMS          rfg14jun85)
1 5 probs epiderm start" hair"      head: class.parms
2 s" bilateral" mark s" scales" 10 %, s" exoskeleton" 10 %, t: plhi @ 1+ record# off
3 s" skin" 5 %, s" hair" 5 %, s" feathered surface" 5 %,    begin whichguy record# @ over <
4 s" armor" 5 %, s" surface" 10 %, s" patterned surface" 7 %, while shaper pelts
5 s" fur" 10 %, -1. %remain %,      shape 1.5@ exterior 1.5@
6 s" radial" mark s" scales" 10 %, s" exoskeleton" 10 %,    clhi @ 1+ cllo @ 1+ 2dup - \ more than one order?
7 s" fur" 10 %, s" skin" 5 %, s" hair" 5 %, s" armor" 5 %, if do i record# ! whichguy
8 s" surface" 10 %, s" patterned surface" 7 %, -1. %remain %, 2dup exterior 1.5! 2over shape 1.5!
9                                     loop
10 s" spherical" mark                else 2drop
11 s" scales" 10 %, s" fur" 0 %, s" hair" 5 %, s" armor" 5 %, then 2drop 2drop 1 record# +!
12 s" surface" 10 %, -1. %remain %,    repeat drop t;
13 s" irregular" mark s" surface" 100 %,
14 s" amorphous" mark s" surface" 100 %,
15 defaultstart

```

30

```

0 \ probs arrays for animal orders ground appendages rfg05jun84
1
2 5 probs animal.appendage
3 start" cilia"
4 s" bilateral" mark 0. 20 %, s" leg" 40 %,
5 s" tentacle" 30 %, s" pseudopod" 5 %, s" cilia" %remain %,
6 s" radial" mark 0. 5 %, s" leg" 30 %,
7 s" tentacle" 40 %, s" pseudopod" 20 %, s" cilia" %remain %,
8 s" spherical" mark 0. 40 %, s" leg" 5 %,
9 s" tentacle" 5 %, s" pseudopod" 30 %, s" cilia" %remain %,
10 s" amorphous" mark 0. 20 %, s" leg" 5 %,
11 s" tentacle" 5 %, s" pseudopod" 40 %, s" cilia" %remain %,
12 s" irregular" mark 0. 20 %, s" leg" 5 %,
13 s" tentacle" 5 %, s" pseudopod" 30 %, s" cilia" %remain %,
14 defaultstart
15

```

31

```

0 \ textures for surfaces
1 9 probs appearance defaultstart
2 s" surface" maybe start" soft"
3 s" waxy" Op s" downy" Op s" filmy" Op s" dull" Op
4 s" shivering" Op s" metallic" Op s" serrated" Op s" bristly" Op
5 s" fuzzy" Op s" hairy" Op s" warty" Op s" rubbery" Op
6 s" plastic-like" Op s" hard" Op s" fleshy" Op s" leathery" Op
7 s" spongy" Op s" fibrous" Op s" oily" Op s" thorny" Op
8 s" veined" Op s" pocked" Op s" bumpy" Op s" jagged" Op
9 s" dripping" Op s" cracked" Op s" shiney" Op
10 s" smooth" Op s" sticky" Op s" gooey" Op s" gummy" Op
11 s" dry" Op s" horny" Op s" mucousy" Op s" soft" Op
12 s" glassy" Op s" crystalline" Op s" gnarly" Op =chance
13 defaultstart
14
15

```

32

```

0 \ continue surface textures
1 s" skin" maybe s" waxy" Op s" filmy" Op s" dull" Op
2 s" leafy" Op s" mucousy" Op s" hairy" Op s" warty" Op
3 s" horny" Op s" plastic-like" Op s" hard" Op s" fleshy" Op
4 s" dry" Op s" spongy" Op s" oily" Op s" veined" Op s" pocked" Op
5 s" coarse" Op s" bumpy" Op s" crusty" Op s" slick" Op
6 s" cracked" Op s" shiney" Op s" wet" Op s" rough" Op s" soft" Op
7 s" dripping" Op s" sticky" Op s" gooey" Op s" gummy" Op
8 s" chitinous" Op s" leathery" Op s" rubbery" Op =chance
9 s" hair" maybe s" flowing" Op s" wirey" Op s" tangled" Op
10 s" matted" Op s" bristly" Op s" fuzzy" Op s" oily" Op
11 s" curly" Op s" shiney" Op s" smooth" Op s" soft" Op
12 s" coarse" Op =chance
13 s" fur" maybe s" wirey" Op s" tangled" Op s" curly" Op
14 s" matted" Op s" bristly" Op s" fuzzy" Op s" oily" Op
15 s" coarse" Op s" shiney" Op s" smooth" Op s" soft" Op =chance

```

33

```

s" feathered surface" maybe s" curly" Op s" matted" Op
start" soft" s" curly" Op s" matted" Op
s" downy" Op s" bristly" Op s" bristly" Op s" fuzzy" Op
s" fleshy" Op s" leathery" Op s" coarse" Op s" shiney" Op
s" soft" Op s" leafy" Op =chance defaultstart
s" scales" maybe
start" soft" s" filmy" Op s" dull" Op s" boney" Op
s" chitinous" Op s" leafy" Op s" serrated" Op s" fuzzy" Op
s" rubbery" Op s" horny" Op s" plastic-like" Op s" hard" Op
s" fleshy" Op s" leathery" Op s" dry" Op s" spongy" Op
s" oily" Op s" gnarly" Op s" striated" Op s" bumpy" Op
s" crusty" Op s" slick" Op s" jagged" Op s" glassy" Op
s" shiney" Op s" crystalline" Op s" rough" Op
s" soft" Op s" smooth" Op =chance defaultstart

```

34

```

0 \ continue (patch) surface textures
1 s" armor" maybe
start" soft" s" filmy" Op s" dull" Op s" boney" Op
s" chitinous" Op s" horny" Op s" plastic-like" Op
s" hard" Op s" leathery" Op s" dry" Op s" thorny" Op
s" gnarly" Op s" striated" Op s" pocked" Op
s" bumpy" Op s" crusty" Op s" shiney" Op s" crystalline" Op
s" rocky" Op s" rough" Op s" smooth" Op =chance
defaultstart

```

35

```

0 \ finish surface textures
1 s" exoskeleton" maybe
start" soft" s" filmy" Op s" dull" Op s" boney" Op
s" chitinous" Op s" horny" Op s" plastic-like" Op
s" hard" Op s" leathery" Op s" dry" Op s" thorny" Op
s" gnarly" Op s" striated" Op s" pocked" Op s" bumpy" Op
s" crusty" Op s" shiney" Op s" crystalline" Op
s" rough" Op s" smooth" Op =chance defaultstart
s" patterned surface" mark 0. 100 %,

```

36

```

0 \ weapon fetature probability array
1 2 probs weaponry
2 animals 0 mark
3 0. 30 %, s" mandible" 5 %, s" teeth" 5 %,
4 s" beak" 5 %, s" claws" 5 %, s" quills" 5 %, s" pincer" 5 %,
5 s" tusk" 5 %, s" horn" 5 %, s" stinger" 5 %, s" poison sac" 5 %,
6 s" acid sac" %remain %,
7 plants 0 mark
8 0. 20 %, s" thorns" 30 %, s" poison sac" 20 %,
9 s" acid sac" %remain %,

```

10

11

12

13

14

15

37

```

0 \ decorative feature probability array
1 2 probs decorations
2 plants 0 mark
3 s" protrusion" 5 %, s" flowers" 15 %, s" fruits" 4 %,
4 s" berries" 4 %, s" projection" 5 %, s" bark" 4 %,
5 s" bulbs" 5 %, s" leaves" 20 %, s" vines" 10 %,
6 s" stem" 10 %, s" branches" %remain %,
7 animals 0 mark
8 s" snout" 5 %, s" feeler" 3 %, s" shell" 5 %,
9 s" protrusion" 4 %, s" crest" 2 %, s" mane" 2 %,
10 s" ear" 7 %, s" compound eye" 9 %, s" eye spot" 7 %,
11 s" eye stalk" 7 %, s" eye" 10 %, s" head" 9 %,
12 s" mouth" 5 %, s" whiskers" 3 %, s" plumes" 2 %,
13 s" projection" 4 %, s" tata" 1 %, s" hooter" 1 %,
14 s" bazwango" 1 %, s" fendibular extremulator" 3 %,
15 s" tail" %remain %,

```

38

```

0 \ #of... legs
1 5 probs #legs s" bilateral" mark
2 1. 10 %, 1. 5 %, 2. 18 %, 4. 10 %, 6. 12 %, 8. 12 %, 12. 10 %,
3 3. 1 %, 5. 1 %, 7. 3 %, 9. 3 %, 11. 3 %, 10. %remain %,
4 s" radial" mark
5 2. 2 %, 4. 2 %, 6. 12 %, 8. 12 %, 10. 12 %, 12. 2 %,
6 1. 2 %, 3. 2 %, 5. 2 %, 7. 12 %, 9. 12 %, 11. 12 %,
7 32767. %remain %, \ this represents 'many' in this instance
8 s" spherical" mark
9 0. 30 %, 2. 10 %, 4. 2 %, 6. 5 %, 8. 5 %, 10. 5 %, 12. 2 %,
10 1. 2 %, 3. 2 %, 5. 2 %, 7. 5 %, 9. 5 %, 11. 5 %,
11 32767. %remain %,
12 s" irregular" mark 0. 20 %, 6. 4 %, 7. 4 %, 8. 4 %, 9. 4 %,
13 10. 4 %, 11. 4 %, 32767. %remain %,
14 s" amorphous" mark 0. 100 %,
15

```

39

```

\ movement mode for beings with legs
start" sessile"
6 probs legg \ indexed by gravity
1. mark s" walking" 5 %, s" hopping" 10 %,
s" jumping" 30 %, s" leaping" 55 %,
2. mark s" walking" 10 %, s" hopping" 15 %,
s" jumping" 30 %, s" leaping" 45 %,
3. mark s" walking" 15 %, s" hopping" 20 %,
s" jumping" 30 %, s" leaping" 35 %,
4. mark s" walking" 35 %, s" hopping" 30 %,
s" jumping" 20 %, s" leaping" 15 %,
5. mark s" walking" 45 %, s" hopping" 30 %,
s" jumping" 15 %, s" leaping" 10 %,
6. mark s" walking" 55 %, s" hopping" 30 %,
s" jumping" 10 %, s" leaping" 5 %,
defaultstart

```

rfg08jun8

40

```

\ orders sizing and setting class fields
head: sizing
t: record# @ >r (planet) 1.5@
>c+s plan-mass @ 100 / 5 min 1+ iclose
6 swap - 20 * peak ! 10 100 slip! 12 / 1+ 1 max 9 min
r> record# ! whichguy size.index c! t;

head: spec>color
t: rput >r >r (system) 1.5@ >c+s @inst-species
ecase ascii M eof ( red) 1 endof ascii K eof ( orange) 9 endof
ascii G eof ( yellow) 11 endof ascii F eof ( white) 24 endof
ascii A eof ( green) 15 endof ascii B eof ( blue) 19 endof
ascii 0 eof ( very blue) 27 endof
abort" Spectral class not found" endcase iclose
r> r> rget t;

```

rfg18apr85

41

```

\ color selection forall plants orders
head: chloro \ lo,hi ---
t: record# @ >r 2dup rrnd dup
r@ record# ! whichguy reddest c! swap 1+ 27 min
rrnd r> record# ! whichguy bluest c! drop t;

head: phyll \ runs before kingdoms
t: spec>color ( yellowish green) 13 over <
if drop ( brownish green) 14
( purple) 22
else ( yellow) 11 <
if ( red) 1
( yellowish orange) 10
else ( copper-colored) 8
( blue) 19 then then t;

```

rfg20may84



42

```

0 \ color selection forall plants orders
1
2 head: dye
3 t: rput >r >r niche c@ (kng) animals =
4   if ( red) 1 ( black) 26
5   else coin
6     if ( multi-colored) 23
7       ( black) 26
8   else phyll
9   then
10  then r> r> rget whichguy chloro t;
11
12 head: ?pure \ color object --- object,tflag if pure color
13 t: rput >r >r 2dup >c+s \ color object --- object,fla
14   @inst-species 7 = iclose r> r> rget t;
15

```

43

```

0 \ select weapon
1
2 head: fortify
3 t: rand% niche c@ (kng) 0 weaponry result 2dup or
4   if ?s if 255
5     else ? if 0
6       else 1
7       then
8       then
9       else dup
10      then w# c! w.append 1.5! t;
11
12 head: decorate
13 t: rand% niche c@ (kng) 0 decorations result
14   d.append 1.5! t;
15

```

44

```

0 \ animate #of legs
1 decimal
2 head: trimming
3 t: d.append 1.5@ 2dup or
4   if .?
5     if 1
6     else ?s 0=
7     if rand% shape 1.5@ #legs result drop
8     dup 16 / 0= if 1+ then
9     else 0
10    then
11    then
12    else 0
13    then d# c! 2dup t;
14
15

```

45

```

rfg20may8 \ flying generation rfg14jun85) 4
%% grounded 30 , 10 , 0 , -30 , -100 , -100 ,
%% bouyant -50 , 0 , 20 , 40 , 50 ,
v: .fly v: .large
head: 5'0 t: 0 0 0 0 t;
head: %fly \ --- %flying,size,index
t: record# @ >r
  (planet) 1.5@ >c+s plan-mass @ 100 / 5 min 1+
  dup >r grounded atmo.density c@ iclose
  dup >r bouyant +
  0 100 constrain
  r> r> - 5 + dup * 9 /
  r> record# ! whichguy t;
head: ?large t: size.index c@ .large @ < t;
head: ?fly t: .fly @ rand% < t;

```

46

```

\ continue setting flying orders rfg04jul84
head: elevate \ call only after ground appendage set
t: record# @ >r %fly .large ! .fly !
  niche c@ (kng) animals = ?large and
  if .fly @ rand% <
  if rand% 80 <
    if s" wing" s" flying"
    else s" float sac" s" floating" then
    coin 1+ 2*
    else 5'0 then
  else ?large if ?fly if coin if s" float sac" s" floating" 1
  else 5'0 then else 5'0 then else 5'0 then
  then r> record# ! whichguy
  a# c! a.mobility 1.5! a.append 1.5! t;

```

47

```

rfg19jun8 \ plant mobility rfg14jun85) 4
v: %mobile
head: ?mobile \ --- [%mobile] for plants; all animals move
t: plants @ dup
  if drop 30 peak ! 0 40 slip! \ this is the 'y' value
  25 +
  then t; \ leaves either 0 or some useable percentage

\ remember: peak is really scaled 256 100 */ (still??)

```

48

```

0 \ continue with plants/animals
1
2 head: nubile
3 t: ?mobile %mobile !
4   niche c@ (kng) plants =
5   if %mobile @ niche c@ (producer) 02 =
6     if 25 - 0 max then
7       rand% >
8       if s" mobile"
9       else s" sessile"
10      then
11      else s" mobile"
12      then g.mobility 1.5! t;
13
14
15

```

49

```

0 \ choose animals orders ground appendage (and plants) rfg05jun8
1
2 head: walk \ [order] --- sets ground appendage
3 t:  rand% shape 1.5@
4   animal.appendage result t;
5
6 head: stroll
7 t:  niche c@ (kng) animals =
8   if walk
9   else g.mobility 1.5@ s" mobile" d=
10   if rand% 35 <
11     if s" sessile" g.mobility 1.5! 0 0
12     else s" tentacle" then
13     else 0 0 then
14   then whichguy g.append 1.5! t;
15

```

50

```

0 \ choosing ground movement mode
1
2 head: ?move \ [ground appendage] --- [mobility mode]
3 t:  ecase s" tentacle" 2of s" crawling" endof
4     s" pseudopod" 2of s" oozing" endof
5     s" cilia" 2of s" creeping" endof
6     s" leg" 2of rand% (planet) 1.5@ >c+s
7   plan-mass @ 100 / 5 min 1+ iclose 0 legg result endof
8     0 0 2of shape 1.5@ 2dup s" spherical" d=
9   if 2drop s" rolling" else s" bilateral" d=
10   if s" slithering" else s" moving" then then endof
11   abort" something wrong in ?move " endcase t; exit
12
13
14
15

```

51

```

rfg04jun84 \ setting ground movement mode
head: movea t: g# c@ 0= if s" moving" else
      g.append 1.5@ ?move then t;

head: movep t: g.mobility 1.5@ s" mobile" d=
      if s" crawling" else s" sessile" then t;

head: movem t: record# @ >r
      niche c@ (kng) animals =
      if movea else movep then
      r> record# ! whichguy g.mobility 1.5! t;

```

rfg16may85)

52

```

\ animate #of legs
head: animate
t:  g.append 1.5@ >r >r i i' or
if i i' ?s not
  if ?
    if 255
      else rand% shape 1.5@ #legs result drop
      then
    else 255
      then >r 2drop r> dup g# c!
      0= if 0 0 g.append 1.5! then
      then r> r> 2drop t;

```

rfg28jul8

53

rfg08jun84 \ finish surface textures

```

head: texturize
t:  rand% exterior 1.5@
      appearance result texture 1.5! t;

```

54

57

```

0 \ copying order information: stack, retrieve and drfg14jun85) \ BIO module caller
1
2 head: do.oparms : bio plhi off ov-cancel
3 t: sizing dye fortify decorate trimming dio save-buffers ov-cancel
4 elevate nubile stroll animate movem texturize t; plhi @
5 if c+oparms ov-cancel then ;
6
7
8
9
10
11
12
13
14
15

```

55

```

0 \ copying minimal information: regions and niches rfg14jun85)
1 head: order.parms
2 t: plhi @ 1+ record# off
3 begin whichguy record# @ over <
4 while do.oparms
5 orhi @ 1+ record# !
6 repeat drop t;
7 : c+oparms class.parms order.parms ;
8
9 exit
10 cr ." ova= " ova ?
11 cr ." t-dp= " t-dp ?
12 cr ." +2= " t-dp 2+ ?
13 cr ." +4= " t-dp 4 + ?
14 cr ." here= " here u.
15

```

56

```

0 \ OVERLAY SUFFIX: SETTING CLASS/ORDER AFIELDS rfg14jun85)
1 trace @ trace off dispose trace !
2 close-overlay
3 89 overlay hp-ov
4 hp-ov
5
6 forth definitions
7 : c+oparms \ sets class and order creatures afields
8 hp-ov hiparms c+oparms
9 save-buffers ov-cancel ;
10
11 ov-cancel save-buffers
12 31 width !
13
14
15

```