

```

0 \ OVERLAY PREFIX: TV INTERFACE AND BEHAVIOR
1 \ phrased getter/debugger
2 \ start of phrase to iaddr retrieval
3 \ localized search
4 \ scanning word for finding a particular phrase
5 \ actual s* word compiles double literal instance
6 \ at sea test for wandering
7 \ start of alien lifeform simulation: fields, prob
8 \ conds: ?player-firing-laser to ?alien-hit-with-l
9 \ conds: ?random-angry to ?stunned
10 \ conds: niche testing
11 \ conds: ?next-to-mate to ...
12 \ conds: ?aggr-enough to ?mate-just-eaten
13 \ conds: ?out-of-bounds to ?stuns>stun-points
14 \ conds: ?moving to ?wants-to-float
15 \ conds: ?wants-to-land to ?ok-to-land
16 \ delta-xy computes what to add to reduce/increas
17 \ ?move-.icon if it is visible
18 \ macros using delta-xy
19 \ actions: reduce-aggr-level to attack-player
20 \ actions: flee-player to flee-its-mate
21 \ actions: attack-its-mate to become-unpaired
22 \ actions: wander to reduce-hits-by-1-to-40
23 \ actions: reduce-.by- to increment-stuns
24 \ actions: fly to land
25 \ simulate does it for each creature
26 exit \ search-list for a possible mate
27 exit \ pairing
28 \ quick: post players vitality
29 \ attack player routine
30 \ attack player routine
31 \ window erase for the current window
32 \ attack player routine
33 \ combining attack routines
34 \ expert: alien conditions --> actions
35 \ expert: conditions --> actions ( towards player)
36 \ rules: responses to other lifeforms c --> a
37 \ rules: responses to other lifeforms c --> a
38 \ rules: responses to other lifeforms c --> a
39 \ rules: responses to other lifeforms c --> a
40 \ rules: responses to other lifeforms c --> a
41 \ rules: responses to other lifeforms c --> a
42 \ rules: responses to other lifeforms c --> a
43 \ rules: responses to other lifeforms c --> a
44 \ rules: responses to other lifeforms c --> a
45 \ rules: responses to other lifeforms c --> a
46 \ rules: responses to other lifeforms c --> a
47 \ rules: responses to other lifeforms c --> a
48 \ rules: air - ground transitions
49 \ rules: air - ground transitions
50 \ simula wraps the experts up
51 \ simulate does it for each creature
52 \ .stardate
53 \ store stuff
54 ( CASE STATEMENT BY CHARLES EAKER
55 \ 'angle' to text phrase case statement
56 \ button to action case statement
57 \ BLTs for 'function keys'
58 \ case for do buttons
59 \ tv-key
60 \ .distance
61 \ .cargo % full
62 \ .energy (other stuff) for terrain vehicle
63 \ absolute>degrees conversion for display
64 \ ?forced+hr , module callers
65 \ tv-tasks
66 \ tv-untasks
67 \ mainline routine
68 \ OVERLAY SUFFIX: behave-ov
69 \ continue overlay suffix: TV word

```

0

3

```

0 \ OVERLAY PREFIX: TV INTERFACE AND BEHAVIOR      (rfq06sep85) \ localized search
1                                                    \ still uses ?first exit test, so doesn't wrap
2 vocabulary behavior immediate                    2v= (Starting)
3 115 open-overlay
4 behavior definitions                             : defaultStart
5                                                    file# @ record# @ >r >r
6 2290 trans-allot                               present ci (starting) 1.5! iclose iclose
7 newt-dp                                          r> r> record# ! file# ! ;
8 9 width !                                       defaultstart
9
10                                                    : startAt
11                                                    box-iaddr 1.5@ >c (starting) 1.5@ >c+s ;
12
13
14
15

```

1

4

```

0 \ phrased getter/debugger
1                                                    \ scanning word for finding a particular phrase
2 head: 72dup t: 2dup or if 2dup then t;          \ follow s" with the phrase object to be found
3 head: difind t: 2dup ifind t;                   : (s") 0 dup found 1.5!
4 head: 1.5, t: here 3 allot 1.5! t;              record# @ >r file# @ >r
5                                                    'nop 'map ! ' inext 'travers ! ' ?>first 'exit !
6 head: PRESENT t: BOX-IADDR 1.5@ >c+s IOPEN ifirst t; startat scan"
7                                                    begin ?match inext ?first found 1.5@ or or until
8 head: tvwindow                                  found 1.5@ - 0=
9 t: 64 3 7 38 window -1 wbottom +! t;            if ." Phrase " type ." not found" abort then
10                                                    2drop iclose iclose found 1.5@
11 head: coin t: 0 2 rrnd t;                       r> file# ! r> record# ! ;
12 head: rand% t: 0 100 rrnd t;
13 head: >flag t: 0= 0= t; \ ensures that result of AND op is a fl
14
15

```

2

5

```

0 \ start of phrase to iaddr retrieval
1 transient trace @ trace off
2 2V= found
3
4 : scan" \ parse to " --- addr,count
5  ascii " word count 2dup >uppercase ;
6
7 : ?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

```

```

\ actual s" word compiles double literal instance rfq20may85)

: s" (s") [compile] 2literal ; immediate
: start" [compile] s" (starting) 1.5! ;
resident trace !

head: tvxy
t: tvehicle 1.5@ >c+s
inst-x @ inst-y @ iclose t;
( xabs @ yabs @ ? )
head: its-xy t: inst-x @ inst-y @ t;

```

6

```

0 \ at sea test for wandering          rfg21jun85)
1 \ should also be applied to approach, but too difficult
2 ( right now )
3
4 head: ?in-ocean \ --- tflag 1 if in ocean
5 t: inst-x @ xcon @ - inst-y @ ycon @ -
6   acelladdr a@ 32 < t;
7
8
9
10
11
12
13
14
15

```

7

```

0 \ start of alien lifeform simulation: fields, prob(rfg06sep85)
1 68 54 1 afield: i.level  68 55 1 afield: a.level
2 68 48 1 afield: slowest  68 51 1 afield: stun.effect
3 68 12 1 afield: niche    68 41 3 afield: a.mobility
4 68 17 1 ifield: hits     68 18 1 ifield: stuns
5 68 52 1 afield: hit.points
6 68 24 3 ifield: goal-instance
7 68 23 1 ifield: goal-direct
8 68 21 1 ifield: move-mode 68 19 1 ifield: aggression
9 68 27 1 ifield: behave
10 68 11 1 afield: size.index
11 42 17 1 ifield: tv-weapon
12 11 17 2 ifield: content-vol
13 32 17 1 afield: atmo.activity
14
15

```

8

```

0 \ conds: ?player-firing-laser to ?alien-hit-with-1(rfg07oct85)
1 : ?player-firing-laser  weapon-fired @
2 tvehicle 1.5@ >c+s tv-weapon c@ 1 = iclose and ;
3 : ?alien-close-enough
4   its-xy eyexy @ - abs 6 < >r
5   eyexy 2+ @ - abs 6 < >r and ;
6
7 : ?alien-smart-enough i.level c@ 69 > ;
8 head: ?ultra-aggr t: aggression 79 > t;
9
10 : ?random-scared rand% 10 < ;
11
12 : ?alien-hit-with-laser its-xy eyexy d@ d=
13 tvehicle 1.5@ >c+s tv-weapon c@ 1 = iclose and
14   dup if 32 behave c@ or behave c! then ;
15

```

9

```

\ conds: ?random-angry to ?stunned          (rfg07oct85)
head: ?random-angry t: rand% 10 < t;

: ?alien-hit-with-stunner its-xy eyexy d@ d=
tvehicle 1.5@ >c+s tv-weapon c@ 0= iclose and ;

head: ?mobile t: slowest c@ >flag t; \ or test for g.mobility<>s

: ?notice-player behave c@ 64 and 0=
  if rand% 10 < if behave c@ 64 or behave c! 1 else 0 then
  else 1 then ; \ make sure logic is right
: ?next-to-player
tvxy its-xy rot - abs 2 < >r - abs 2 < >r and ;

head: ?alive t: hits c@ >flag t;
head: ?stunned t: stuns c@ 0> not t;

```

10

```

\ conds: niche testing          rfg29aug85)
head: ?carnivore t: niche c@ 8 and >flag t;
head: ?herbivore t: niche c@ 4 and >flag ?carnivore not and t;
\ : ?omnivore ?carnivore niche c@ 4 and >flag and ;
head: ?producer t: niche c@ 2 and >flag ?herbivore not and
      ?carnivore not and t;
head: get-mate
t: goal-instance 1.5@ >c+s t;

\ : ?mate-carnivore get-mate ?carnivore iclose ;
\ : ?mate-herbivore get-mate ?herbivore iclose ;
\ : ?mate-omnivore get-mate ?omnivore iclose ;
\ : ?mate-producer get-mate ?producer iclose ;

```

11

```

\ conds: ?next-to-mate to ...
\ : ?next-to-mate its-xy get-mate its-xy iclose
\ rot - abs 2 < >r - abs 2 < >r and ;
\ : ?mate-on-ground get-mate move-mode c@ 0= iclose ;

\ : ?has-mate goal-instance 1.5@ or >flag ;

\ : ?mate-alive
\   get-mate hits c@ >flag iclose ;
head: ?big-enough t: size.index c@ get-mate
      size.index iclose - 1 > t;
\ : ?mate-approaching get-mate goal-direct c@ iclose ;
\
\ : ?mate-attacking get-mate behave c@ 1 and iclose ;

```

12

```

0 \ conds: ?aggr-enough to ?mate-just-eaten
1
2 head: ?aggr-enough t: aggression c@ 39 > t;
3
4 head: ?can-break-away t: rand% 10 < t;
5
6 head: ?attacked t: behave c@ 128 and >flag t;
7
8 head: ?attacking t: behave c@ 1 and t;
9
10 : ?mate-out-of-bounds 0 ;
11
12 : ?randomly-unpaired coin ;
13
14 \ : ?mate-just-eaten get-mate behave c@ 8 and >flag iclose ;
15

```

13

```

0 \ conds: ?out-of-bounds to ?stuns>stun-points rfg14aug85)
1
2 head: ?out-of-bounds t: false t;
3
4 head: ?been-eaten t: behave c@ 8 and >flag t;
5
6 head: ?hits>0 t: hits c@ 0 > t;
7
8 head: ?stuns>0 t: stuns c@ 0 > t;
9
10 head: ?stuns<0 t: stuns c@ 0 < t;
11
12 : ?stuns(stun-points stuns c@ stun.effect c@ < ;
13
14 head: ?HITS=0 t: HITS C@ 0= t;
15

```

14

```

0 \ conds: ?moving to ?wants-to-float rfg29aug85)
1
2 head: ?moving t: behave c@ 16 and >flag t;
3
4 head: ?on-ground t: move-mode c@ 0= t;
5
6 head: ?can-fly t: a.mobility 1.5@ s" flying" d= t;
7
8 head: ?can-float t: a.mobility 1.5@ s" floating" d= t;
9
10 : ?wants-to-fly rand% 70 < ;
11
12 : ?wants-to-float ?wants-to-fly ;
13
14
15

```

15

```

\ conds: ?wants-to-land to ?ok-to-land rfg29aug85)
head: ?wants-to-land t: rand% 30 < ?attacking or t;
head: ?wandering t: behave c@ 2 and >flag t;
head: ?ok-to-land t: its-xy xcon @ - swap ycon @ -
    acelladdr a@ 16 > t;
head: ?dumb-enough t: i.level c@ 35 < t;
head: ?dangerous t: aggression c@ 65 > t;
head: ?wimpy t: aggression c@ 50 ( 35) < t;

```

16

```

\ delta-xy computes what to add to reduce/increasrfg21jun85)
\ x,y are the moving coordinates, x1,y1 are fixed
head: delta-xy \ x,y x1,y1 --- x,y deltax,deltay
t: 2over 2over d=
    if 2drop -1 -1 \ force movement in case same points
    else 2over rot swap - dup
        if dup abs / then >r
        - dup
        if dup abs / then r>
    then t;
head: move-yes t: behave c@ 16 or behave c! t;
head: move-no t: behave c@ 16 negate or behave c! t;

```

17

```

\ ?move-.icon if it is visible (rfg07oct85)
head: ?move-.icon \ move only visible icons
t: xormode @ xormode off
ci ?icon=iaddr drop \ assume in list
point>icon
inst-x @ inst-y @ ( 2dup )
!iy !ix
>mainview .background \ plot only if visible
.local-icons v>display
>display xormode ! t;
head: reorg \ force reorganization of iconlist
t: -5000 dup anchor 2! orglist t;

```


18

```

0 \ macros using delta-xy                                rfg14aug85)
1
2 head: mate-xy \ --- x,y of goal-instance ( GI(>0!!)
3 t: goal-instance 1.5@ >c+s
4 inst-x @ inst-y @ iclose t;
5
6 head: approach \ x,y xl,yl --- ! move xy towards xyl
7 t: delta-xy rot + inst-y ! + inst-x !
8 ?move-icon t;
9
10 head: flee \ xy,xyl --- ! what it sez...
11 t: delta-xy negate rot + inst-y ! negate + inst-x !
12 ?move-icon t;
13
14
15

```

19

```

0 \ actions: reduce-aggr-level to attack-player rfg29aug85)
1
2 : reduce-aggr-level
3 a.level c@ 1 51 rrnd - 0 max a.level c! ;
4
5 : increases-aggr-level
6 a.level c@ 1 21 rrnd + 100 min a.level c! ;
7
8 head: approach-player
9 t: behave c@ 1 or behave c!
10 its-xy tvxy approach move-yes t;
11
12 head: flee-player
13 t: its-xy tvxy flee move-yes t;
14
15

```

20

```

0 \ actions: flee-player to flee-its-mate rfg21jun85)
1
2 head: approach-its-mate
3 t: its-xy mate-xy approach 1 goal-direct c! move-yes t;
4
5 : flee-its-mate
6 its-xy mate-xy flee 0 goal-direct c! move-yes ;
7
8
9
10
11
12
13
14
15

```

21

```

\ actions: attack-its-mate to become-unpaired rfg29aug85)
head: ?dead t: hits c@ 0= \ change blt if creature's dead
if ci ?icon=iaddr drop point>icon dead-ic !ic then t;

\ head: attack-its-mate t: 16 negate behave c@ or
\ 1 or behave c! t;

\ head: eat-its-mate t: rand% 10 <
\ if get-mate @ behave c@ or behave c! then iclose t;
\ should also unlink mate from display??

head: become-unpaired
t: goal-instance 1.5@ 0. goal-instance 1.5!
c) 2swap >c+s 0. goal-instance 1.5!
cdrop >c+s t;

```

22

```

\ actions: wander to reduce-hits-by-1-to-40 rfg19aug85)

head: wander t: \ stuns c@ stun.effect c@ < not
( if ) behave c@ 2 or 254 and behave c! \ not attacking
inst-x @ inst-y @ -1 2 rrnd inst-x +! -1 2 rrnd inst-y +!
?in-ocean if inst-y ! inst-x ! else 2drop then
?move-icon move-yes ( then ) t;

: erase-lifeform-from-array
-icon idelete reorg ;

: reduce-hits-by-10-to-50
hits c@ 10 51 rrnd - 0 max hits c! ?dead ;

: reduce-hits-by-1-to-40
hits c@ 1 41 rrnd - 0 max hits c! ?dead ;

```

23

```

\ actions: reduce..by- to increment-stuns rfg22may85)
head: change-icon t: -icon icon-parm module +icon reorg t;

: reduce-stuns-by-10-to-50
stuns c@ 10 51 rrnd - ( 0 max ) stuns c! ;

: stun-effects-hits
hits c@ stuns c@ abs - 0 max hits c! 0 stuns c! ;

head: die t: move-no 0 hits c! 0 move-mode c! t;
head: stunned t: 0 stuns c! t;
head: unstunned t: stun.effect c@ stuns c! t;

: increment-stuns
1 stuns c@ + stun.effect c@ min stuns c! ;

```

24

```

0 \ actions: fly to land
1
2
3 head: fly t: 1 move-mode c! change-icon ?move-.icon t;
4
5 head: float t: 2 move-mode c! change-icon ?move-.icon t;
6
7 head: land t: 0 move-mode c! change-icon ?move-.icon t;
8
9
10
11
12
13
14
15

```

27

```

exit \ pairing
head: form-pairs \ within ilocal display list
t: iindex @ >r \ preserve before searching
ilocal @ 0 do i point>icon
    @id 19 <
    if @il @ih >c+s
        goal-instance 1.5@ or 0= \ living only
        if search-list 2dup or 0= not \ not paired?
            if (pair)
                else 2drop leave \ find one
            then
        then iclose
    then loop
r> point>icon t;

```

25

```

0 \ simulate does it for each creature
1
2 head: this-creature \ ( --- current creature) for task
3 t: ^crit @ point>icon
4 @il @ih >c+s t;
5
6 head: 1st-creature t: 0 ^crit ! t;
7
8
9
10
11
12
13
14
15

```

28

```

\ quick: post players vitality
17 ( assign-crew) 17 3 ifield: captain
16 31 1 ifield: ^vit

```

26

```

0 exit \ search-list for a possible mate
1
2 head: search-list \ --- 0. or iaddr, mates for CI
3 t: 0. ilocal @ iindex @ 1+ \ unsuccessful flag on stack
4 do i point>icon \ examine each icon as a possible mate
5 @id 19 <
6 if @il @ih >c+s
7 goal-instance 1.5@ 2dup or
8 if 2swap leave then 2drop iclose \ lose 0.
9 then loop t;
10
11 head: (pair) \ mate2 --- (mate1 --- mate2) ! iaddr's
12 t: 2dup goal-instance 1.5! \ forms a pair
13 c> 2swap >c+s
14 goal-instance 1.5! t;
15

```

29

```

rfg21jun8 \ attack player routine
head: @attacks
t: rand% dup 71 <
if drop 1
else 95 <
if 2 else 3 then
then t;
head: lcrew \ sets ci to a random crewmember
t: *assign-crew >c+s captain
0 6 rrnd 3 * + 1.5@ iclose t;

```

30

```

0 \ attack player routine
1 16 ( crewmember) 11 15 ifield: ^name
2 16 19 1 afield: ^dur
3
4 head: ?success \ --- true if succesful, else false
5 t: rand%
6 ?flat @ if 10 * then
7 aggression c@ i.level c@ + 2/ < t;
8
9 head: damage \ --- value! amount of damage to crewmember
10 t: aggression c@ hit.points c@ 2* + 3 / t;
11
12
13
14
15

```

31

```

0 \ window erase for the current window
1
2 head: werase \ using current window
3 t: color @
4 wtop @ wleft @ wbottom @ 1- wright @ 1+
5 black poly-window-fill
6 !color t;
7
8 head: ?injure \ --- flag true if in bad storm and should injur
9 t: #storm @ 6 11 within
10 if 1 100 rrnd 30 <
11 if 'injure @ module then
12 then t;
13
14
15

```

32

```

0 \ attack player routine
1
2 head: .attack \ ( creature --- crewmember)
3 t: tvwindow werase
4 wleft @ wtop @ 4 - pos. white !color
5 ." THE " resembles 1.5@ 2dup or 0=
6 if 2drop shape 1.5@ then >c+s
7 phrase phr-cnt c@ type cdrop iclose
8 ." LIFEFORM ATTACKS "
9 wleft @ wtop @ 10 - pos. ( BCR ?)
10 >c+s ^name $. t;
11
12
13
14
15

```

33

```

rfq29may85) \ combining attack routines rfg03jul85)
head: attack-player
t: behave c@ 1 or behave c! \ set attacking bit
#attacks 1 \ this number of attacks
do ci ?success damage
!crew .attack
swap if ^dur c@ / ^vit c@ swap -
0 max 100 min ^vit c!
?heal on
else drop ." , BUT MISSES"
then ." ." iclose >c+s 500 ms \ read pause
loop move-no
'death @ execute
'.vital-signs @ execute t;

```

34

```

(rfg11sep85) \ expert: alien conditions --> actions
9 3 expert Aggr-modifier
rule: ?player-firing-laser true ?alien-close-enough true
?alien-smart-enough true ?ultra-aggr false
?random-scared true --> reduce-aggr-level
rule: ?alien-hit-with-laser true ?ultra-aggr true
?random-angry true --> increases-aggr-level
rule: ?alien-hit-with-stunner true ?ultra-aggr true
?random-angry true --> increases-aggr-level

```

35

```

\ expert: conditions --> actions ( towards player)rfq03may85)
7 3 expert vs.player
rule: ?mobile true ?alive true ?stunned false
?notice-player true ?dangerous true
?next-to-player false --> approach-player
rule: ?alive true ?stunned false ?notice-player true
?dangerous true ?next-to-player true
--> attack-player
rule: ?mobile true ?notice-player true ?alive true
?stunned false ?wimpy true --> flee-player

```

36

```

0 \ rules: responses to other lifeforms c --> a
1 5 50 expert life-sim
2 exit
3 rule: ?mobile true ?alive true ?stunned false
4     ?herbivore true ?mate-producer true
5     ?next-to-mate false ?mate-on-ground true
6     ?has-mate true --> approach-its-mate
7
8 rule: ?mobile true ?alive true ?stunned false
9     ?carnivore true ?mate-herbivore true
10    ?big-enough true ?next-to-mate false
11    ?mate-on-ground true ?has-mate true
12    --> approach-its-mate
13
14
15

```

37

```

0 \ rules: responses to other lifeforms c --> a
1 exit
2 rule: ?mobile true ?alive true ?stunned true
3     ?omnivore true ?mate-producer true
4     ?next-to-mate false ?mate-on-ground true
5     ?has-mate true --> approach-its-mate
6
7 rule: ?mobile true ?alive true ?stunned false
8     ?omnivore true ?mate-herbivore true
9     ?big-enough true ?next-to-mate false
10    ?mate-on-ground true ?has-mate true
11    --> approach-its-mate
12
13
14
15

```

38

```

0 \ rules: responses to other lifeforms c --> a
1 exit
2 rule: ?mobile true ?alive true ?stunned false
3     ?herbivore false ?producer false
4     ?mate-herbivore false ?mate-producer false
5     ?big-enough true ?ultra-aggr true
6     ?next-to-mate false ?mate-on-ground true
7     ?has-mate true --> approach-its-mate
8
9 rule: ?mobile true ?alive true ?stunned false
10    ?mate-approaching true ?has-mate true
11    --> flee-its-mate
12
13
14
15

```

39

```

\ rules: responses to other lifeforms c --> a
exit
rule: ?mobile true ?alive true ?stunned false
     ?mate-attacking true ?aggr-enough false
     ?can-break-away true ?has-mate true
     --> flee-its-mate

rule: ?alive true ?stunned false ?herbivore true
     ?mate-producer true ?next-to-mate true
     ?mate-alive true ?has-mate true
     --> attack-its-mate

```

40

```

\ rules: responses to other lifeforms c --> a
exit
rule: ?alive true ?stunned false ?carnivore true
     ?mate-herbivore true ?big-enough true
     ?next-to-mate true ?mate-alive true
     ?has-mate true --> attack-its-mate

rule: ?alive true ?stunned false ?omnivore true
     ?mate-producer true ?next-to-mate true
     ?mate-alive true ?has-mate true
     --> attack-its-mate

```

41

```

\ rules: responses to other lifeforms c --> a
exit
rule: ?alive true ?stunned false ?omnivore true
     ?mate-herbivore true ?big-enough true
     ?next-to-mate true ?mate-alive true
     ?has-mate true --> attack-its-mate

rule: ?alive true ?stunned false ?herbivore false
     ?producer false ?mate-herbivore false
     ?mate-producer false ?big-enough true
     ?ultra-aggr true ?next-to-mate true
     ?mate-alive true ?has-mate true
     --> attack-its-mate

```


42

```

0 \ rules: responses to other lifeforms c --> a
1 exit
2 rule: ?alive true ?stunned false ?mate-attacking true
3     ?aggr-enough true ?has-mate true
4     --> attack-its-mate
5
6 rule: ?alive true ?stunned false ?carnivore false
7     ?producer false ?mate-producer true
8     ?attacked true ?mate-attacking false
9     ?next-to-mate true ?has-mate true
10    --> eat-its-mate
11
12
13
14
15

```

43

```

0 \ rules: responses to other lifeforms c --> a    rfg03jul85)
1 exit
2 rule: ?alive true ?stunned false ?carnivore false
3     ?producer false ?mate-producer true
4     ?mate-alive false ?next-to-mate true
5     ?has-mate true --> eat-its-mate
6
7 rule: ?alive true ?stunned false ?producer false
8     ?herbivore false ?mate-producer false
9     ?mate-alive false ?next-to-mate true
10    ?has-mate true --> eat-its-mate
11
12
13
14
15

```

44

```

0 \ rules: responses to other lifeforms c --> a
1
2 \ rule: ?has-mate true ?mate-out-of-bounds true
3 \     --> become-unpaired
4
5 \ rule: ?has-mate true ?randomly-unpaired true
6 \     --> become-unpaired
7
8 \ rule: ?has-mate true ?mate-just-eaten true
9 \     --> become-unpaired
10
11 rule: ?alive true ?stunned false ?mobile true
12     --> wander
13
14
15

```

45

```

\ rules: responses to other lifeforms c --> a
2 2 expert erase-life

rule: ?out-of-bounds true
    --> erase-lifeform-from-array

rule: ?been-eaten true
    --> erase-lifeform-from-array

```

46

```

\ rules: responses to other lifeforms c --> a    rfg21aug85)
5 3 expert ?reduce-points

rule: ?alien-hit-with-laser true ?hits>0 true
    --> reduce-hits-by-10-to-50

\ rule: ?mate-attacking true ?hits>0 true
\     --> reduce-hits-by-1-to-40

rule: ?alien-hit-with-stunner true ?stuns>0 true
    --> reduce-stuns-by-10-to-50

head: ?stuns>hits t: ?stuns<0 if stun-effects-hits then t:

```

47

```

\ rules: responses to other lifeforms c --> a    rfg21jun85)
54 3 expert effect-life

rule: ?hits=0 TRUE ?alive true --> die

rule: ?stuns>0 false ?stunned false --> stunned

rule: ?stuns>0 true ?stunned true --> unstunned

head: ?bump-stuns
t:    ?alive ?stunned and
      ?stuns<stun-points and
      if increment-stuns then t:

```

48

```

0 \ rules: air - ground transitions      rfg29aug85)
1 14 7 expert ground>air
2
3 rule: ?moving true ?on-ground true ?can-fly true
4       ?alive true ?wants-to-fly true --> fly
5
6 rule: ?moving true ?on-ground true ?can-float true
7       ?alive true ?producer true --> float
8
9 rule: ?moving true ?on-ground true ?can-float true
10      ?producer false ?wants-to-float true
11      ?alive true --> float
12
13
14
15

```

49

```

0 \ rules: air - ground transitions      rfg10aug85)
1
2 rule: ?alive false ?on-ground false
3       ?ok-to-land true --> land
4
5 rule: ?stunned true ?on-ground false
6       ?ok-to-land true --> land
7
8 rule: ?on-ground false ?wandering true ?wants-to-land true
9       ?ok-to-land true --> land
10
11 rule: ?on-ground false ?attacking true
12       ?ok-to-land true --> land
13
14
15

```

50

```

0 \ simula wraps the experts up          (rfg02sep85)
1
2 head: (simulate)
3 t: weapon-fired @
4 if ' ?reduce-points distract ?reduce-points drop then
5   hits c@
6 if ' aggr-modifier distract aggr-modifier drop then
7   ' vs.player distract vs.player 0=
8   if ' life-sim distract life-sim drop then
9     ' erase-life distract erase-life drop
10  ?stuns>hits
11  ' effect-life distract effect-life drop
12  ?bump-stuns
13  ' ground>air distract ground>air drop t;
14
15

```

51

```

\ simulate does it for each creature      (rfg10sep85)
: simulate
?new @                                     \ set by landing or TVMOVE
if ( form-pairs ) 1st-creature ?new off ( maybe also +icon?? )
then ilocal @ 2 > \ anybody to behave
if this-creature @id 19 <
  if (simulate)
    weapon-fired dup @ \ set by firing
    if -1 swap +! else drop then \ each creature exam once
    then iclose \ the displayed object
\ iindex @ 0= if form-pairs then
^crit @ 1+ ilocal @ 1- mod ^crit ! \ wrap to 0 if at end
then ;

```

52

```

\ .stardate                                (rfg12sep85)
head: .stardate \ print current stardate
t: xormode @ xormode off
    116 178 pos. 10 black poly-erase-text
white !color stardate @
300 /mod 4620 + swap 30 /mod 1+ swap 1+
dup 10 < if ." 0" then 0 .r ." -"
dup 10 < if ." 0" then 0 .r ." -" 0 .r xormode ! t;

```

53

```

\ storm stuff
create %storm \ probability table, 0<=index<=4
0 , 10 ( 3) , 20 ( 6) , 40 ( 15) , 60 ( 30) ,
\ index is atmo.activity

: ?fair storm @ 0= atmo @ 0= 0= and
if 1 4 rrnd %storm ! 100 %eff !
else atmo @ 0= if 0 %storm ! 100 %eff ! then then ;

: ?storm storm @ 0= atmo @ 0= 0= and
if (planet) 1.5@ >c+s atmo.activity c@ iclose
dup + %storm + @ rand% >
if 'storm @ module then \ calls forward patch
else storm @ 1- 0 max storm !
then ?fair ?injure ; \ fair weather phrase, maybe injure

```

54

```

0 ( CASE STATEMENT BY CHARLES EAKER          rfg01jul85)
1 DECIMAL transient trace @ trace off
2 : 20F          4 ?PAIRS COMPILE 20VER COMPILE d= COMPILE
3              0BRANCH HERE 0 , COMPILE 2DROP 5 ; IMMEDIATE
4 : !CSP SP@ CSP ! ;
5 : ECASE        ?COMP CSP @ !CSP 4 ; IMMEDIATE
6
7 : ENDOF        5 ?PAIRS COMPILE BRANCH HERE 0 ,
8              SWAP 2 [COMPILE] THEN 4 ; IMMEDIATE
9
10 : ENDCASE      4 ?PAIRS COMPILE DROP BEGIN SP@
11              CSP @ = 0= WHILE 2 [COMPILE]
12              THEN REPEAT CSP ! ; IMMEDIATE
13 resident trace !
14
15

```

55

```

0 \ 'angle' to text phrase case statement      rfg12apr85
1 head: angle>text \ deltax,delay --- prints direction
2 t: ecase
3 0 1 2of ." NORTH" endof
4 1 1 2of ." NORTHEAST" endof
5 1 0 2of ." EAST" endof
6 1 -1 2of ." SOUTHEAST" endof
7 0 -1 2of ." SOUTH" endof
8 -1 -1 2of ." SOUTHWEST" endof
9 -1 0 2of ." WEST" endof
10 -1 1 2of ." NORTHWEST" endof
11 drop ( extra parameter ) endcase t;
12
13
14
15

```

56

```

0 \ button to action case statement            (rfg12sep85)
1 head: icons+map t: xormode off
2 >mainview .background .local-icons v>display >display t;
3
4 head: ?embark \ --- true! if tv back on ship icon, after moved
5 t: (ship) 1.5@ >c+s inst-x @ inst-y @ iclose
6     tvxy d= ( moved @ and ) t; \ on top of tv
7
8
9
10
11
12
13
14
15

```

57

```

\ BLTs for 'function keys'          rfg08jul85)
create functions
6 c, 12 c, 2 c, 12 c, 20 c, 12 c, 22 c, 12 c, 20 c,
  12 c, 42 c, 12 c, 20 c, 12 c, 62 c, 12 c, 20 c,
  12 c, 82 c, 12 c, 28 c, 12 c, 110 c, 12 c, 24 c,
  12 c, 134 c, 12 c, 23 c,

head: set-cursor
t: tv>window functions set-crs drop t;

head: get-distance \ --- tv distance from ship : ( --- ship )
t: (ship) 1.5@ >c+s
tvxy inst-y @ - dup * 0
rot inst-x @ - dup * 0 d+ sqrt t;

```

58

```

\ case for do buttons                (rfg11sep85)
v: .stats ( forward patch for icons )
: maps ' maps module ;
: walk&talk ' (talk) module ;
: tvmove ' tvmove module time-passing on ;
: do.weapons ' (weapons) module ;
: /items ' /items module ;
: icons ' icons module ' (tvset) module
  .stardate .stats @ execute
  .vital-signs @ module ; \ maybe lucky this works

case >buttons
0 is maps 1 is tvmove 2 is walk&talk 3 is walk&talk
4 is do.weapons 5 is /items 6 is icons
others nop

```

59

```

\ tv-key                                (rfg10sep85)
head: Movecrs t: ncrs +! set-cursor t;

head: tv-key \ key --- !terrain vehicle console interaction
t: (xyscan) ecase 0 1 2of 1 movecrs endof \ right
  0 -1 2of -1 movecrs endof \ left
  drop ( extra parameter )
  ?trig if ncrs @ >buttons \ execute this button
  icons+map \ refresh display
  then endcase
?embark if 0= \ flip task active flag
-1 dup yvis +! yabs +! time-passing off
then t; \ fix ship position, prevent display

```

60

```

0 \ .distance
1 17 23 3 ifield navig 16 27 1 ifield ^nav
2 : -storm storm @ 0= ;
3 : ?nav \ --- flag! true if good navigator
4 *assign-crew >c+s navig 1.5@ >c+s ^nav c@
5 cdrop iclose 199 > ;
6
7 : .distance \ prints KM. from ship + direction
8 116 150 pos. 4 black poly-erase-text white !color
9 -storm ?nav or
10 if get-distance 4 .r ." KM." then \ ( --- ship )
11 116 143 pos. 09 black poly-erase-text white !color
12 -storm ?nav or
13 if tvxy inst-x @ inst-y @ iclose \ { ship --- }
14 delta-xy 2swap 2drop angle>text then ; \ prints direction
15

```

61

```

0 \ .cargo % full
1
2 head: .cargo
3 t: 116 157 pos. 3 black poly-erase-text
4 tv-hold 1.5@ >c+s
5 content-vol @ dup
6 if 5 max 5 / then \ inst-qty =500 for tv-hold
7 white !color 3 .r ." % FULL"
8 iclose t;
9
10
11
12
13
14
15

```

62

```

0 \ .energy (other stuff) for terrain vehicle (rfg12sep85)
1
2 head: .energy \ print % of full tank
3 t: 116 171 pos. 7 black poly-erase-text
4 tvehicle 1.5@ >c set-current
5 inst-qty @ iclose dup 0> not \ get amt in tank (full=32767)
6 if drop 'ending @ execute \ head back to starport
7 else ( 116 171 pos. ) white !color
8 2000 - 100 30767 */ dup 0> not \ is it zero percent
9 if drop color @ pink !color ." RESERVE" !color
10 else 3 .r ." %"
11 then 116 164 pos. 3 black poly-erase-text
12 %eff @ ( 48 ) 100 e/m @ negate */
13 99 min 3 .r ." %"
14 then t;
15

```

63

```

(rfg06sep85) \ absolute>degrees conversion for display
head: abs>deg \ x,y --- xdegree,ydegree
t: 480 - 10 53 */ swap
1152 - 10 64 */ swap t;

head: ?east \ xdegree --- print E if negative, else W
t: ?dup if 0< if ." E" else ." W" then
else ." "
then t;

head: ?south \ ydegree --- print S if negative, else N
t: ?dup if 0< if ." S" else ." N" then
else ." "
then t;

```

64

```

(rfg06sep85) \ ?forced+hr , module callers (rfg11sep85)
head: .where
t: -storm ?nav or
if white !color tvxy abs>deg 16 197 pos.
dup dup 0< if negate then 4 .r ?south ." * "
dup dup 0< if negate then 0 .r ?east then t;

head: .stats
t: xormode @ >r xormode off .where
.energy .cargo .distance r> xormode ! t;
'.stats '.stats !
head: ?forced+hr \ --- flag! true if force display of stats
t: ?new-hour forced @ or forced off t;

\ head: .vits t: ' (.vits) module t;

```

65

```

(rfg12sep85) \ tv-tasks (rfg11sep85)
: tv-tasks \ install vectors for tasker
' ?forced+hr 'vehicle-cycle ! ' ?storm 'repair !
'.stats 'vehicle-status ! \ vehicle update, every hour
'obits 'death ! \ executes when there is a death
'simulate 'external-events ! \ lifeform simulation goes here
'beep 'ending ! \ KLUDGE for testing endgame=no fuel
'?appoint 'crew-cycle ! 0. lastappoint 2!
'.vits 'vital-signs ! ' heal 'treatment ! ?heal on

'.stardate '.date ! \ date in auxwindow
time-passing on \ activate these routines
forced on ; \ print vehicle stats

```


66

69

```

0 \ tv-untasks                                (rfq10sep85) \ continue overlay suffix: TV word      (rfq08sep85)
1
2 head: tv-untasks \ remove vectors for tasker                                : tv \ call terrain vehicle interaction
3 t: ' ?ship-repair 'vehicle-cycle ! ' mrepair 'repair !                    ' heaveho 'throw-away !
4 ' nop 'vehicle-status ! \ vehicle update, every hour                        behave-ov behavior tv ;
5 ' obits 'death !
6 ' nop 'external-events ! \ lifeform simulation went here                    : (simula) behave-ov behavior simulate ;
7 ' ?appoint 'crew-cycle !                                                    : simulate ' (simula) module ;
8 ' nop 'vital-signs ! ' heal 'treatment !                                    ' simulate 'simulate ! \ so can be called from MOVE tasker
9 ' nop 'date ! t; \ date in auxwindow                                         ov-cancel
10
11 head: tv-cleanup t: ' (clean) module tv-untasks t;
12
13
14
15

```

67

```

0 \ mainline routine                        (rfq11sep85)
1
2 head: role-call t: ' (role) module t;
3
4 head: tv-init
5     t: ' tv-disp module ,stardate role-call
6     tv-tasks black lcolor ! t;
7
8 : tv ' tv-cleanup ' tv-key ' tv-init dotasks
9     time-passing on ;
10
11
12
13
14
15

```

68

```

0 \ OVERLAY SUFFIX: behave-ov                (rfq08sep85)
1
2 trace @ trace off dispose trace !
3
4 close-overlay 115 overlay behave-ov behave-ov
5
6 forth definitions \ available to module callers
7
8 : heaveho ' (heave) module ;
9
10 : tvtasks behave-ov behavior tv-tasks ;
11 ' tvtasks 'tv ! \ for move-ov
12
13
14
15

```