

```

0 ( overlay prefix-----COMBAT-----
1 \ basic routines
2 \ instance and species variables: VESSEL
3 \ instance and species variables: SHIP
4 \ crew variables
5 \ dist
6 \ delta-xy
7 \ cases to orientations
8 \ conditions: ?attacking, ?start-attacking-hostile
9 \ ?has-missiles ?has-lasers
10 \ ?oh to fire, ?fire laser now
11 \ ?terminated ?go-now ?can-surrender
12 \ ?nothing-happening
13 \ ?stop-attacking ?move-rate>0
14 \ ?range-long
15
16 \ ?in-combat ?p-has/missiles/lasers
17 \ posture stuff
18 \ ?move-rnd-now
19 \ refresh auxilliary screen
20 \ missile record
21 \ missile array
22 \ (.missiles)
23 \ missile DDA
24 \ step-dda showspace
25
26 \ missile utilities- find, install
27 \ missile utilities- find, install
28 \ ?hit
29 \ ?any-hits
30
31 \ ?move-icon if it is visible
32 \ macros using delta-xy
33 \ find-bomb
34 \ ?plasma, advance
35 \ all-repair for test
36 \ attack simulation
37 \ orient current vessel
38 \ display start
39 \ install-icons, reorient
40 \ init-space-map
41 \ 4:lscale 2:lscale
42 \ 8map
43 \ dist-test finds greatest distance
44 \ dist-test finds greatest distance
45 \ determine which scale to use
46 \ space-disp
47 \ fire-missile
48 \ fire-laser and execute alien
49 \ flee, approach
50 \ >approaching move-randomly
51 exit \ raise-shields, lower-shields
52 exit \ arm-weapons, disarm-weapons
53 \ combat expert
54 \ more rules
55 \ yet more rules

```

```

55 \ yet more rules
56 \ and yet more rules
57 \ simulation loop
58 \ simulate does it for each creature
59 \ simula wraps the experts up
60 \ simulate does it for each creature
61 \ probe offsets
62 \ compute xy of probe
63 \ check if player has mounted weapons
64 \ find-fuel
65 \ deplete energy for using armaments
66 \ ?empty-tank save-fuel
67 \ compute nav skill dithering
68 \ compute target coordinates, given probe
69 \ fire missile from our side
70 \ fire-laser and execute alien
71 \ pchoose - laser or missile ( if got )
72 \ combat-tasks
73 \ tv-untasks
74 \ tv-key
75 \ combat-init
76 \ war
77 ( overlay suffix-----COMBAT-----
78 \ module callers for combat

```

0

3

```

0 ( overlay prefix-----COMBAT----- (rfg03oct85) \ instance and species variables: SHIP (rfg13oct85).
1 20 c: sship
2 9 width ! sship 17 1 ifield: %hull-pts sship 18 1 ifield: %armor-clc
3 vocabulary combat-voc immediate sship 19 1 ifield: %engine-pts sship 20 1 ifield: %engine-clc
4 132 open-overlay sship 21 1 ifield: %sensor-pts sship 22 1 ifield: %sensor-clc
5 combat-voc definitions sship 23 1 ifield: %comm-pts sship 24 1 ifield: %comm-clc
6 ( sp0 @ ova ! ) sship 25 1 ifield: %shield-pts sship 26 1 ifield: %shield-clc
7 1600 trans-allot newt-dp sship 27 1 ifield: %missile-pts sship 28 1 ifield: %missile-clc
8 sship 29 1 ifield: %laser-pts sship 30 1 ifield: %laser-clc
9 sship 48 2 ifield: shield-cnt sship 69 4 ifield: %status
10 sship 50 2 ifield: armor-cnt
11 sship 73 1 ifield: %heading
12
13
14
15

```

1

4

```

0 \ basic routines (rfg11oct85) \ crew variables (rfg06oct85)
1 head: >flag t: 0= 0= t; \ insures 0bit set
2 17 23 3 ifield: navigator
3 head: rand% t: 1 100 rrnd t; 16 27 1 ifield: ^nav
4 16 31 1 ifield: ^vit \ crewmember vitality
5 head: ship-coords \ --- x,y 16 19 1 afield: ^dur
6 t: xabs @ yabs @ t;
7
8 head: ves-coords \ --- x,y | assume pointed to ci vessel icon
9 t: @ix @iy t;
10
11 head: @xy t: inst-x @ inst-y @ t;
12
13 head: ship>c t: *ship >c+s t;
14
15

```

2

5

```

0 \ instance and species variables: VESSEL (rfg17oct85) \ dist (rfg25sep85)
1 decimal 25 c: vessel
2 2v: probe \ holds xy of probe point
3 vessel 11 1 afield: ves-firert v: tdist
4 \ vessel 12 2 afield: ves-armorpts 2v: target
5 vessel 14 2 afield: ves-shieldpts v= ?combat \ not used
6 vessel 18 1 afield: ves-missile v= ?attack
7 vessel 19 1 afield: ves-laser
8 vessel 20 1 afield: ves-plasma
9 vessel 23 4 ifield: ves-last-fired
10 vessel 12 1 ifield: ves-speed
11 vessel 18 2 ifield: ves-armor-hits
12 vessel 20 2 ifield: ves-shield-hits
13 vessel 17 1 ifield: ves-behave
14 vessel 11 1 ifield: ves-heading
15

```

head: dist \ x,y,x1,y1 --- distant  
t: rot - dup \* >r  
- dup \* 0 r> 0 d+ sqrt t;

head: @rclass \ --- resource class  
t: c@ 7 and t;

6

9

```

0 \ delta-xy (rfg11oct85) \ ?has-missiles ?has-lasers (rfg17oct85)
1 \ x,y are the moving coordinates, x1,y1 are fixed
2
3 : rdit begin -1 2 rrnd ?dup until ; head: ?has-lasers
4 t: ves-laser c@ >flag t;
5 : delta-xy \ x,y x1,y1 --- x,y deltax,deltay
6 2over 2over d= head: ?has-plasma
7 if 2drop 0. \ rdit rdit \ force movement in case same points t: ves-plasma c@ >flag t;
8 else 2over rot swap - dup
9 if dup abs / then >r head: ?has-missiles
10 - dup t: ves-missile c@ >flag ?has-plasma or t;
11 if dup abs / then r>
12 then ;
13
14
15

```

7

10

```

0 \ cases to orientations (rfg11oct85) \ ?oh to fire, ?fire laser now (rfg01oct85)
1 head: save-face : sometime ;
2 t: @inst-class 25 = if ves-heading \ vessel orientation
3 else %heading
4 then c@ t;
5
6 case x=-1
7 ( y= ) -1 is 5 1 is 3 0 is 4 others unravel head: ?fire-miss-now
8 case x=0 t: ves-firert c@ 10 * rand% < t;
9 ( y= ) -1 is 6 1 is 2 0 is save-face others unravel
10 case x=1
11 ( y= ) -1 is 7 1 is 1 0 is 0 others unravel head: ?fire-laser-now
12 t: ves-firert c@ 5 * rand% < t;
13 case key>orient
14 ( x= ) -1 is x=-1 0 is x=0 1 is x=1 others unravel
15

```

8

11

```

0 \ conditions: ?attacking, ?start-attacking-hostile(rfg26sep85) \ ?terminated ?go-now ?can-surrender (rfg09oct85)
1 52 12 1 ifield: csurrender \ can surrender flag
2 head: ?attacking
3 t: ?attack @ >flag head: ?terminated t: terminated @ >flag t;
4 a-posture @ 4 > and dup ?attack ! t;
5
6 head: ?start-attacking-hostile
7 t: rand% 20 < t; head: ?go-now t: rand% 70 < t;
8
9
10
11 head: ?can-surrender
12 t: (aoriginator) 1.5@ >c+s csurrender c@ iclose t;
13
14
15

```

12

15

```

0 \ ?nothing-happening (rfg06oct85)
1 120000. 2c= boredom
2 2v= excite \ time lasttime something happened
3
4 : ?nothing-happening
5   time d@ boredom excite d@ d+ d) ;
6
7
8 : !time
9   time d@ excite d! ;
10
11 \ talkcount is a global variable, modified by communications
12 \ set to some initial-value, it counts down every time
13 \ external-events executes (i guess)
14
15

```

13

16

```

0 \ ?stop-attacking ?move-rate>0 (rfg19sep85) \ ?in-combat ?p-has/missiles/lasers (rfg13oct85).
1
2 head: ?stop-attacking \ : ?in-combat ?combat @ >flag ;
3 t: rand% 5 < t; \ ?combat on
4
5 head: ?p-has-missiles
6 head: ?move-rate>0 t: *ship >c+s %missile-cls @rclass >flag
7 t: ves-speed c@ >flag t; %missile-pts c@ >flag and iclose t;
8
9 head: ?p-has-lasers
10 t: *ship >c+s %laser-cls @rclass >flag
11 %laser-pts c@ >flag and iclose t;
12
13
14
15

```

14

17

```

0 \ ?range-long (rfg06oct85) \ posture stuff (rfg01oct85)
1 11 c: long-distance 5 c: short-distance
2 head: ?obsequious t: a-posture c@ 1 = t;
3 head: ?range \ distance --- flag ! true if within distance
4 t: >r @xy head: ?hostile t: a-posture c@ 8 = t;
5 ship-coords head: ?fight t: a-posture c@ 16 = t;
6 rot - abs r@ < >r
7 - abs r> swap r> < and t;
8 \ a-posture is a global, handled by communications
9 head: ?range-long t: long-distance ?range not t;
10 head: ?range-short t: short-distance ?range t;
11 head: ?range-medium t: ?range-long not ?range-short not and t;
12 \ should agree with VISWINDOW
13 \ may recode to speed access ( keep object on stack)
14 \ rather than space savings
15

```

18

```

0 \ ?move-rnd-now
1 v: toggler toggler off
2
3 head: ?move-rnd-now
4 t: rand% 30 <
5 if toggler -1 toggle then
6 toggler @ t;
7
8 \ flips bit occasionally
9
10
11
12
13
14
15

```

21

```

(rfg06oct85) \ missile array (rfg23sep85)
create marray maxmis 1+ missize * allot \ Missile array
\ the right way would be with linked lists, ...
\ must use save-overlay for this to work
\ maybe put missile stuff in separate overlay, for tasker speed

head: mclr
t: marray maxmis missize * 0 fill t; mclr

: >m \ missile# ---! sets MISSILE
missize * marray + missile ! ;

: mcoords currx @ curry @ ;
: mdest destx @ desty @ ;

```

19

```

0 \ refresh auxilliary screen
1
2 ' ovdbars c: #ov/damage
3 ' ov/status c: #ov/status
4 ' nop c: #nop
5
6 case #aux>module
7 1 is #ov/status 3 is #ov/damage others #nop
8
9 : refresh-aux #aux @ #aux>module module ;
10
11
12
13
14
15

```

22

```

(rfg06oct85) \ (.missiles) (rfg22oct85)
head: 2dot t: ." ." t;
head: 4dot t: >2font ." ." >1font t;
head: 8dot t: currx @ 2 mod if ." +" else ." *" then t;

case scale>miss
2 is 2dot 4 is 4dot 8 is 8dot others nop

: (.missiles)
white !color maxmis 0 do i >m @mclass \
if mcoords wld>scr scr>blt
swap 4 + swap pos.
xwld:xpix @ scale>miss
then loop ;

```

20

```

0 \ missile record (rfg12oct85)
1 v= missile \ holds address of current missile record
2 10 c= maxmis 22 c: missize
3 1 c: #player 0 c: #alien
4 head: misf
5 t: create c, ( c: ? )
6 does> c@ missile @ + t; \ returns address in MARRAY
7
8 0 misf currx 4 misf curry \ current location,
9 8 misf destx 10 misf desty \ destination
10 12 misf morig \ originator 1 =Player, 0=Alien
11 13 misf mclass \ holds weapon class
12 14 misf deltax 18 misf deltax \ increment for dda
13
14 : @mclass mclass c@ 15 and ;
15

```

23

```

\ missile DDA (rfg11oct85)
head: compd \ test delta, max --- newdelta ( signed double )
t: >r 0 swap 1 r> m*/ t;

head: set-dda
t: destx @ currx @ - dup abs
desty @ curry @ - dup abs rot max >r
r@ compd deltax d!
r> compd deltax d!
32768 dup currx 2+ ! curry 2+ ! t;

```



24

```

0 \ step-dda showspace
1
2 head: step-dda
3 t: currx d@ deltax d@ d+ currx d!
4 currx d@ deltax d@ d+ currx d! t;
5
6 : showspace
7 >mainview dark (.missiles)
8 .local-icons v>display >display ;
9 \ should reposition near beginning of file, then substitute
10
11
12
13
14
15

```

25

```

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

```

26

```

0 \ missile utilities- find, install (rfq11oct85)
1 v: #missiles \ can we create more missiles? Max of 9
2 #missiles off
3
4 v: #explode \ # of explosions this iteration
5 #explode off
6 \ clear after servicing explosion
7
8 head: mfind \ x,y --- addr ! 0,0=empty slot, or currentx
9 t: missile @ >r \ save during search
10 maxmis 0
11 do i >m 2dup mcoords d=
12 if leave then loop 2drop \ coordinates
13 missile @ r> missile ! t; \ leaves addr in missile array
14
15

```

27

```

\ missile utilities- find, install (rfq11oct85)
head: minstall \ source x,y,dest x,y,owner,class ---
t: #missiles @ maxmis < \ locked out at higher level by ?fire-now
if 0. mfind missile ! mclass c! morig c!
desty ! destx ! currx ! currx !
set-dda 1 #missiles +! ( INSTALL DDA PARMS HERE )
else 2drop 2drop 2drop
then t; \ sets MISSILE if space found
\ might need success flag

head: mdelete \ --- ! remove MISSILE from array, fails if not in
t: missile @ missile 0 fill
-1 #missiles +! -1 #explode +! t;

```

28

```

\ ?hit (rfq22oct85)
: ?hit \ --- flag ! true if hit goal, aliens can't hit self
mcoords \ missile location
ilocal @ ?icons-at dup \ any icons at missile location
if 0 over >r swap \ n,n,0,count
0 do swap point>icon \ examine each icon ( extra work)
morig #malien = \ ?alien missile
if @id 27 35 within \ is icon our ship?
if drop 1 then \ success!
else drop 1 \ our missile, can't hit us (oh?)
then
loop r> >flag and \ merge with icons-at test
then ;
\ doesn't check for class 0 missiles,
\ done in find-bomb and ?any-hits

```

29

```

\ ?any-hits (rfq22oct85)
head: ?any-hits \ check display, see if any collisions
t: maxmis 0 do i >m @mclass \ if any missile in slot
if mcoords ?invis 0= \ off screen
if mdelete 1 #explode +! \ mdelete does -!
else ?hit
if 1 #explode +! then \ record for later processing
then
then loop t;
\ alien missiles always on target?

```

30

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

\ find-bomb

```
: find-bomb \ --- flag! true if one ready; sets MISSLE
#explode @ >r #explode off
maxmis 0 do i >m @mclass
    if ?hit
        if leave then
            then
loop r> #explode ! ; \ use only if certain I exists
```

33

31

0 \ ?move-.icon if it is visible  
1 head: reorg \ force reorganization of iconlist  
2 t: -5000 dup anchor 2! orlist t;  
3  
4 head: ?move-.icon \ move only visible icons  
5 t: xormode @ xormode off  
6 ci ?icon=iaddr drop \ assume in list  
7 point>icon  
8 @xy !iy !ix \ copy from object to icon  
9 reorg showspace xormode ! t;  
10  
11  
12  
13  
14  
15

(rfq22oct85) \ ?plasma, advance

(rfq22oct85)

```
head: ?plasma \ called if plasma bolt has to follow ship
t: @mclass 6 =
    if ship-coords swap destx d! set-dda then t;
\ resets line variables to track our ship

: advance
maxmis 0 do i >m mclass c@
    if ?plasma step-dda
        then loop ;
```

34

32

0 \ macros using delta-xy  
1  
2 head: (movet) \ x,y x1,y1 --- I move xy towards x1y1  
3 t: delta-xy rot + inst-y ! + inst-x ! t;  
4  
5 head: (movea) \ xy,x1y1 --- away  
6 t: delta-xy negate rot + inst-y ! negate + inst-x ! t;  
7  
8  
9  
10 \ seem to be reversed ??  
11  
12  
13  
14  
15

(rfq25sep85) \ all-repair for test

(rfq22oct85)

```
head: heal-crew
t: #assign-crew >c+s inst-y 2+
    dup 18 + swap do
        i 1.5@ >c+s 100 ^vit c! iclose
        3 +loop navigator 1.5@ >c+s 190 ^nav c! iclose iclose t;

head: restore-shield t: 2000 shield-cnt ! 2000 armor-cnt ! t;

: repair-all
ship>c restore-shield heal-crew
100 dup %hull-pts c! dup %engine-pts c!
    dup %sensor-pts c! dup %comm-pts c!
    dup %missile-pts c! dup %laser-pts c!
    %shield-pts c! iclose ;
```

35

36

```

0 \ attack simulation
1
2 : do.missiles
3 advance ?any-hits #explode @ ?dup
4   if 0 do find-bomb
5     mcoords @eclass ?dup 0=
6     if >alpha >0font
7       missile @ marray - missile / ." MISSILE# " . quit
8     then
9       morig c@ 1
10      ' do-damage module mdelete showspace
11    loop then ;
12
13
14
15

```

37

```

0 \ orient current vessel
1
2 head: ?patch \ dx,dy --- flag! true if need to flip orientation
3 t: 2dup or ( neither value is 0 )
4   if / 0> else 2drop 0 then t;
5
6 head: ves-orient \ oldx,y --- ! reorients icon
7 t: @inst-species 25 = not \ debris check
8 if @xy @ 2swap delta-xy \ current location
9   2swap 2drop 2dup
10  key>orient rot rot
11  ?patch if 4 + 8 mod then
12    dup ves-heading c!
13    35 + !id t;
14 then t; \ call before ?move-.icon
15

```

38

```

0 \ display start
1
2 head: orient>ship \ set orientation for attacking
3 t: ship-coords ves-orient
4 ves-heading c@ 4 + 8 mod dup ves-heading c!
5 35 + !id t;
6
7 head: vismap \ sets up x,yvis
8 t: xabs @ 18 - xvis !
9 yabs @ 30 - yvis ! t;
10 \ xvis @ 14 +
11 \ yvis @ 22 +
12 \ xvis @ 22 +
13 \ yvis @ 41 + !viswindow t;
14
15

```

39

```

(rfg22oct85) \ install-icons, reorient

: install-icons \ becomes no-op when finished testing
(encounter) 1.5@ >c+s iopen
' +iconbox module cdrop iclose
*ship ?icon=iaddr
if drop
else *ship >c+s ' icon-param module +icon iclose
then ;

head: reorient \ ship reflects current orientation
t: iindex @ >r
*ship ?icon=iaddr drop point>icon
%heading c@ 27 + !id
r> iindex ! showspace t;

```

40

```

(rfg11oct85) \ init-space-map
(rfg13oct85).

: init-space-map
0. xlldest ! ylldest !
>!iconfont
8 dup xwld:xpix ! ywld:ypix !
1 dup xwld:xpix 2+ ! ywld:ypix 2+ !
vismap ;

```

41

```

(rfg26sep85) \ 4:1scale 2:1scale
(rfg01oct85)

head: 4:1scale
t: vismap >2iconfont
2 dup xlldest ! ylldest !
4 dup xwld:xpix ! ywld:ypix !
1 dup xwld:xpix 2+ ! ywld:ypix 2+ !
xvis @ 10 + yvis @ 15 + xvis @ 26 + yvis @ 43 + !viswindow t;

head: 2:1scale
t: vismap >3iconfont
1 dup xlldest ! ylldest !
2 dup xwld:xpix ! ywld:ypix !
1 dup xwld:xpix 2+ ! ywld:ypix 2+ !
xvis @ yvis @ xvis @ 34 + yvis @ 58 + !viswindow t;

```



42

```

0 \ Bmap
1
2 head: 8:lscale
3 t: vismap >liconfont
4 0. xlldest ! ylldest !
5 8 dup xwld:xpix ! ywld:ypix !
6 1 dup xwld:xpix 2+ ! ywld:ypix 2+ !
7 xvis @ 14 + \ left edge
8 yvis @ 22 + \ bottom
9 xvis @ 22 + \ right edge
10 yvis @ 41 + !viswindow t; \ top
11
12
13
14
15

```

43

```

0 \ dist-test finds greatest distance
1 v= farthest \ largest coordinate gap
2 v= nearest \ closest coordinate gap
3
4 head: (close)
5 t: xabs @ @ix - abs
6 yabs @ @iy - abs min \ largest of two coordinates
7 nearest @ min nearest ! t; \ lowest of stored value and coord
8
9 head: (farth)
10 t: xabs @ @ix - abs
11 yabs @ @iy - abs max \ largest of two coordinates
12 farthest @ max farthest ! t;
13
14
15

```

44

```

0 \ dist-test finds greatest distance
1
2 head: near-far \ set variables for scale and probing
3 t: farthest off 1000 nearest ! \ nothing this far away
4 ilocal @ 0 do
5 i point>icon @id 35 44 within \ is it a vessel?
6 if (close) (farth)
7 then loop t;
8
9
10
11
12
13
14
15

```

45

```

(rfg01oct85) \ determine which scale to use (rfg13oct85).
\ head: getnf t: nearest @ farthest @ t;

: set-scale
( getnf ) near-far
farthest @ dup 5 <
if drop 8:lscale
else 5 8 within ( i guess)
if 4:lscale
else 2:lscale
then
then ( getnf d= not ) l
if showspace then ; \ change display only if change in
\ near/far setting

```

46

```

(rfg02oct85) \ space-disp (rfg13oct85).
: set-orientations
ilocal @ 0 do i point>icon @id 35 44 within
if @il @ih >c+s
ship-coords ves-orient iclose
then loop ;
\ for every ship except ours, set orientation

: space-disp \ sets up space display, assume console in place
\ dclipset
\ init-iconlist init-space-map
( 75 locradius ! install-icons ) reorg
( set-orientations ) set-scale ;
\ showspace ;

```

47

```

(rfg22oct85) \ fire-missile (rfg17oct85)
head: dispoiff
t: 68 + swap 8 + swap t; \ ugh, not scale dependant

: dooff wld>scr dispoiff scr>bit ;

: afire-missile \ from alien to player's ship
ves-coords ship-coords #alien
@inst-species 17 =
if 6
else ves-missile c@
then mininstall step-dda
3 ves-behave c! ?attack on
orient>ship !time ;

```

48

51

```

0 \ fire-laser and execute alien (rfq11oct85) exit \ raise-shields, lower-shields
1 : draw-laser \ x,y,x1,y1,color ---! draws in display
2 >display !color dooff 2swap dooff lline ; : raise-shields ;
3
4 : afire-laser : lower-shields ;
5 ves-laser c@
6 if ves-coords ship-coords red draw-laser
7 " LASER CLASS " white !color .tty
8 ves-laser c@ .
9 #alien 0 ' do-damage module
10 refresh-aux 3 ves-behave c! ?attack on
11 orient>ship !time
12 then ;
13
14
15

```

49

52

```

0 \ flee, approach (rfq13oct85). exit \ arm-weapons, disarm-weapons
1 : coordset ves-coords ship-coords ;
2 : arm-weapons ;
3 : move-away
4 coordset (movea) 4 ves-behave c! : disarm-weapons ;
5 ves-coords ves-orient ?move-icon ; \ flee
6
7 : move-towards \ assumes vessel on c-stack
8 coordset (movet) 3 ves-behave c!
9 ves-coords ves-orient ?move-icon ;
10
11
12
13
14
15

```

50

53

```

0 \ >approaching move-randomly \ combat expert (rfq27sep85)
1
2 head: dit \ --- dither 19 27 expert <combat>
3 t: begin -1 2 rrnd ?dup until t;
4 rule: ?attacking false
5 head: ?approaching ?start-attacking-hostile true ?hostile true
6 t: ves-behave c@ 3 = t; ?has-missiles true --> afire-missile
7
8 head: move-randomly rule: ?attacking false ?fight true
9 t: inst-x @ dit + inst-x ! ?has-missiles true --> afire-missile
10 inst-y @ dit + inst-y !
11 ves-coords ves-orient ?move-icon ; rule: ?attacking true ?has-missiles true
12 ?fire-miss-now true --> afire-missile
13
14
15

```

54

```

0 \ more rules
1
2 rule: ?attacking false
3     ?start-attacking-hostile true ?hostile true
4     ?has-missiles false ?has-lasers true
5     ?range-short true --> afire-laser
6
7 rule: ?attacking false ?fight true
8     ?has-missiles false ?has-lasers true
9     ?range-short true --> afire-laser
10
11
12
13
14
15

```

57

```

(rfg26sep85) \ simulation loop
exit
works like this, i think.
every time through the loop, one of the alien vessels executes
its behavior. Then, all the missile and laser objects are updated

```

55

```

0 \ yet more rules
1
2 rule: ?attacking true ?has-lasers true
3     ?fire-miss-now false
4     ?fire-laser-now true ?range-short true
5     --> afire-laser
6
7 rule: ?hostile false ?obsequious false
8     ?terminated true ?go-now true
9     ?move-rate>0 true --> move-away
10
11 rule: ?obsequious true ?can-surrender false
12     ?move-rate>0 true --> move-away
13
14
15

```

58

```

(rfg26sep85) \ simulate does it for each creature (rfg01oct85)
v= ^ves \ icon# of current vessel for behaving
^ves off

: this-vessel \ ( --- current creature) for task
^ves @ point>icon @ih >c+s ;

head: 1st-vessel t: ^ves off t;

```

56

```

0 \ and yet more rules
1
2 rule: ?nothing-happening true
3     ?move-rate>0 true --> move-away
4
5 rule: ?move-rate>0 true ?obsequious false
6     ?range-long true --> move-towards
7
8 rule: ?move-rate>0 true ?obsequious false
9     ?range-medium true ?has-lasers true
10    ?has-missiles false --> move-towards
11
12 rule: ?move-rate>0 true ?approaching false
13    ?move-rnd-now true --> move-randomly
14
15

```

59

```

(rfg23sep85) \ simula wraps the experts up (rfg12oct85)

head: (combat)
t: ' <combat> distract <combat> drop t;

: fix-ashield \ ( vessel --- vessel ) | charge up shield
ves-shield-hits @ ves-shieldpts @ <
if 5 ves-shieldpts +! then ;

: fix-pshield
shield-cnt @
%shield-cls @rclass \ now depends on shield class
%shield-pts c@ 5 * * < \ and generating capability
if 10 shield-cnt +! then ;

```

60

```

0 \ simulate does it for each creature
1
2 : combat
3 ilocal @ 1 > \ ?anybody to behave
4 if this-vessel @id 35 43 within \ not debris
5   if (combat) fix-ashield
6   else @id 27 35 within \ our ship?
7     if fix-pshield then
8       then iclose \ the displayed object
9       ^ves @ 1+ ilocal @ mod ^ves ! \ wrap to 0 if at end
10 do.missiles near-far set-scale
11 then ;
12
13
14
15

```

63

```

(rfg04nov85). \ check if player has mounted weapons (rfg13oct85).
: ?weap-armed
  %status c@ 1 and ;

```

61

```

0 \ probe offsets
1
2 transient
3 : d, here 4 allot d! ;
4 resident
5
6 create orient>octant \ offsets for probe
7 1 0 d, 1 1 d, 0 1 d, -1 1 d,
8 -1 0 d, -1 -1 d, 0 -1 d, 1 -1 d,
9
10
11
12
13
14
15

```

64

```

(rfg02oct85) \ find-fuel (rfg12oct85)
: find-fuel \ { --- endurium } ! inside starship hold
*starship-hold >c+s iopen
  11 10 ifind drop iopen \ assume elements box
  26 06 ifind drop ; \ also ASSUME some endurium

```

62

```

0 \ compute xy of probe
1
2 : set-probe \ { ship --- ship } ! set xy of probe
3   farthest @ nearest @ + 2/ \ average distance
4   %heading c@ 4* orient>octant + d@
5   >r over * xabs @ +
6   r> rot * yabs @ + probe d! ;
7 exit
8 : show-probe
9   probe d@ wid>scr swap 4 + swap 70 +
10 >display white !color 4 .circle ;
11
12
13
14
15

```

65

```

(rfg03oct85) \ deplete energy for using armaments (rfg12oct85)
2v= #fuel \ hundreths of units of endurium
null #fuel d!

: read-fuel
  find-fuel inst-qty @ 10 m* #fuel d!
  cdrop cdrop iclose ;

: dec-fuel \ amount --- ! to subtract from #fuel
  #fuel d@ rot 0 d-
  2dup null d< if 2drop 0. then
  #fuel d! ;

\ read from hold when combat activated
\ TESTED

```

66

69

```

0 \ ?empty-tank save-fuel (rfq12oct85) \ fire missile from our side (rfq12oct85)
1
2 : ?empty-tank \ --- flag ! true if no fuel left
3 #fuel d@ d0= ;
4
5 : save-fuel \ restore to ship endurium stash
6 #fuel d@ 10 u/mod \ ASSUME no overflow lost!!
7 swap 5 > if 1+ then
8 find-fuel inst-qty !
9 iclose iclose iclose ;
10
11
12 \ check for 0= while running, and restore when finished
13
14
15

```

67

70

```

0 \ compute nav skill dithering (rfq12oct85) \ fire-laser and execute alien (rfq09oct85)
1 2v= dtemp \ dither temps
2
3 : ndither \ navskill --- +/- 5 points
4 0 250 rot
5 - rrnd 50 /
6 0 2 rrnd if negate then ;
7
8 : dither-target \ --- x,ydither for missile to go off
9 *assign-crew >c+s navigator 1.5@
10 >c+s ^nav c@ >r ( 0 to 250 )
11 target 2@ r@ ndither dup abs dtemp ! + swap
12 r> ndither dup abs dtemp 2+ ! + swap
13 cdrop iclose 2dup wld>scr probe d! ( show-probe) ;
14 \ why is wld>scr there :: MAYBE BUG??
15

```

68

71

```

0 \ compute target coordinates, given probe (rfq03oct85) \ pchoose - laser or missile ( if got ) (rfq13oct85).
1
2 : find-target
3 10000 tdist !
4 ilocal @ dup 1 >
5 if 0 do i point>icon @id 35 44 within \ a vessel
6 if probe d@ ves-coords dist
7 dup tdist @ <
8 if tdist ! ves-coords target d!
9 else drop
10 then then loop
11 else drop
12 then ;
13
14
15

```



72

```

0 \ combat-tasks
1 : all-experts 'cex @ execute combat ;
2
3 : combat-tasks \ install vectors for tasker
4 ' ?ship-repair 'vehicle-cycle ! ' Mrepair 'repair !
5 ' nop 'vehicle-status ! \ vehicle update, every hour
6 ' nop 'death ! \ executes when there is a death
7 ' all-experts 'external-events ! \ lifeform simulation goes here
8 ' nop 'ending ! \ KLUDGE for testing endgame=no fuel
9 ' ?appoint 'crew-cycle ! 0. lastappoint 2!
10 ' nop 'vital-signs ! ' heal 'treatment ! ?heal on
11 \ ' nop 'date ! \ date in auxwindow
12 time-passing on ; \ activate these routines
13
14
15

```

73

```

0 \ tv-untasks
1
2 : combat-untasks \ remove vectors for tasker
3 ' ?ship-repair 'vehicle-cycle ! ' Mrepair 'repair !
4 ' nop 'vehicle-status ! \ vehicle update, every hour
5 ' obits 'death !
6 ' CEX+WAX 'external-events ! \ lifeform simulation went here
7 ' ?appoint 'crew-cycle !
8 ' nop 'vital-signs ! ' heal 'treatment !
9 ' nop 'date !
10 save-fuel ?attack off
11 iclose ; \ the ship
12
13
14
15

```

74

```

0 \ tv-key
1 decimal
2 : ?maneuver \ --- flag ! true if space bar hit
3 lkey @ 13 = ;
4
5 : combat-key \ key --- !terrain vehicle console interaction
6 (xyscan) key>orient
7 %heading c! reorient
8 ?trig ?maneuver not and
9 if pfire then showspace
10 ?maneuver if 0= then ; \ leave tasker
11
12
13
14
15

```

75

```

(rfq22oct85) \ combat-init
: combat-init
>mainview dark >display
' (.missiles) '.background !
lst-vessel
' (role) module
combat-tasks !time
read-fuel *ship >c+s
armor-cnt @ 0=
space-disp >!font
if ' ov/damage module 3 else 1 then
  #aux ! refresh-aux
showspace ;

```

76

```

(rfq22oct85) \ war
: war
context-id# @ 4 =
if ' combat-untasks ' combat-key ' combat-init dotasks
else beep then ;

: clear-combat
mclr #missiles off #explode off ;

```

77

```

(rfq30oct85) { overlay suffix-----COMBAT----- (rfq28oct85)
dispose close-overlay
132 overlay combat-ov combat-ov
forth definitions
: (war) combat-voc combat-ov war save-overlay ;

: fixup combat-voc combat-ov repair-all ;

: (comb) combat-voc combat-ov *ship >c+s combat
  iclose save-overlay ;

: cscale combat-voc combat-ov set-scale ;

: cc combat-voc combat-ov clear-combat ;
\ call when leaving encounter

```

78

```
0 \ module callers for combat          (rfg28oct85)
1
2 ov-cancel
3 : combat ' (comb) module ;
4
5 : war ' (war) module ;
6
7 : cb ov-cancel combat-ov ;
8 ' combat 'wax ! ' war 'combat !
9
10 : >w >lores dark ship-console war ;
11
12
13
14
15
```