

```

0 ( SHPMOV - ----- OVERLAY PREFIX -----
1 ( SHPMOV - VAR>C SHP>C PLN>C ORB>C SYS>C ENC>C
2 ( SHPMOV - CONSTANTS
3 ( SHPMOV - VARIABLES
4 ( SHPMOV - INSTANCE FIELDS
5 ( SHPMOV - @XYSHP !XYSHP !%HEADING @%HEADING ?XIDS
6 ( SHPMOV - ?* ?-* ?PLANET ?ENCOUNTER ?SHIP ?ORBIT
7 ( CONTEXT MODULE: PLANET
8 ( Context Module: Planet
9 ( CONTEXT MODULE: PLANET
10 ( Encounter Context ifields
11 ( Encounter Context GET-VESSEL MAKE-VESSELS
12 ( Encounter Context KILL-VESSELS
13 ( Context Module: Aux Graphics P-SURF>COLOR
14 ( Encounter Context .A-VESSEL.AUXENC
15 ( ENCOUNTER CONTEXT ENCSETUP ENCCLEANUP
16 ( Encounter Context ENC-DISP
17 ( Starport Context
18 ( >CONTEXT - MAKE-STAR KILL-STAR
19 ( >CONTEXT - GET-ORBIT#
20 ( >CONTEXT - >ORBIT
21 ( >CONTEXT - ORBIT)
22 ( >CONTEXT - >ORBITS ORBITS> SYSXY
23 ( >CONTEXT - SETRNDXY SETSYSXY
24 ( >CONTEXT - SETRNDXY SETSYSXY
25 ( Context Module: Aux Graphics S-CLASS>COLOR
26 ( CONTEXT MODULE: AUX GRAPHICS .SUN .PLANET
27 ( CONTEXT MODULE: AUX GRAPHICS .ENC .AUXSYS
28 ( >CONTEXT - SYSSETUP SYS-DISP SYSCLEANUP
29 ( >CONTEXT -
30 ( SHPMOV - >[CONTEXT] CTX>C ![CONTEXT]
31 ( SHPMOV - XID-PLAN XID-ORB/*SEC ... ***misc***
32 ( SHPMOV - CONTEXT-CASES RADIUS-CASES ***misc***
33 ( SHPMOV - IVARCASES #CPARMS !CPARMS ***misc***
34 ( SHPMOV - .COORDINATES -ENDURIUM
35 ( Space Dust SDUST >DUST
36 ( Space Dust ?DUST [.DUST]
37 ( SHPMOV - .DUST .MVS
38 ( SHPMOV - ?CHANGEXY .FLUX-ICON
39 ( SHPMOV - >NEWHEADXY
40 ( SHPMOV - #SLOPE #RELXY
41 ( SHPMOV - #QUAD
42 ( SHPMOV - #HEADING
43 ( SHPMOV - HEADDXDY #HEADDXDY #NESTXY
44 ( SHPMOV - #VECTXY MVCLIPSET #DIST
45 ( SHPMOV - #NEWHEADXY #SPEED #BOUNDXY
46 ( SHPMOV - NEST-EFFECTS
47 ( SHPMOV - UNNEST-EFFECTS support
48 ( SHPMOV - NEST-EFFECTS
49 ( SHPMOV - UNNEST-FINIS-MESS
50 ( SHPMOV - UNNEST-START-MESS
51 ( SHPMOV - NEST-FINIS-MESS
52 ( SHPMOV - NEST-FINIS-MESS
53 ( SHPMOV - SEC-DISP DISP-SETUP
54 ( SHPMOV - NEST-SETUP NEST-SHIPXY
55 ( SHPMOV - NEST

```

```

55 ( SHPMOV - NEST
56 ( SHPMOV - RAMVESSEL >ENCOUNTER DOCK ?BOX-NEST
57 ( SHPMOV - >FLUX-NODE
58 ( SHPMOV - COLLIDE
59 ( SHPMOV - ?COLLISION
60 ( SHPMOV - ?DOUBLE-NEST
61 ( SHPMOV - ?STARPORT
62 ( SHPMOV - UNNEST-CLEANUP
63 ( SHPMOV - UNNEST-SHIPXY
64 ( SHPMOV - ?UNNEST ?DOUBLE-UNNEST
65 ( SHPMOV - UNNEST
66 ( SHPMOV - #BOUNDXY AUXLATE AUX-GETTERS
67 ( SHPMOV - FLY
68 ( SHPMOV - SET-DESTINATION
69 ( SHPMOV - SETUP-MOV
70 ( SHPMOV - CLEANUP-MOV
71 ( SHPMOV - COME-HITHER
72 ( SHPMOV INITIALIZERS
73 ( SHPMOV ----- OVERLAY SUFFIX -----

```

0

3

```

0 ( SHPMOV - ----- OVERLAY PREFIX -----10/23/85) ( SHPMOV - VARIABLES 10/17/85)
1 31 width !
2 2V= NAV-TIME
3
4 VOCABULARY SHPMOV-VOC IMMEDIATE
5
6 106 OPEN-OVERLAY
7 SHPMOV-VOC DEFINITIONS
8
9 2600 TRANS-ALLOT
10 NEWT-DP
11
12
13 HEAD: SET-EXTERNAL T: ' !EXT MODULE T;
14
15

```

V: (RADIUS) (radius of current context)
V: [CONTEXT] (context stack pointer to current context)
V: XTRNS V: YTRNS (world x,y coords of nested context)
V: 2NEST 2NEST OFF (double nested context flag)
V: SKIP2NEST SKIP2NEST OFF (force skip over of double nest)
2V: (COLLIDE) (colliding object iaddr) V: XCRS V: YCRS
V: NAV-SKILL (navigator's skill)
V: NAV-CLEAN
V: ?HAS-RING-DEVICE (100% identify flux)
2V: 10*END (10*endurium amt)

1

4

```

0 ( SHPMOV - VAR>C SHP>C PLN>C ORB>C SYS>C ENC>C 3-22-85) ( SHPMOV - INSTANCE FIELDS 6/26/85)
1
2
3 HEAD: VAR>C ( addr -- , ( -- i ) ) T: 1.5@ >C+S T;
4 HEAD: SHP>C ( ( -- i ) \ move ship to context stack)
5 T: (SHIP) VAR>C T;
6 HEAD: PLN>C ( ( -- i ) \ move current planet to context stack)
7 T: (PLANET) VAR>C T;
8 HEAD: ORB>C ( ( -- i ) \ move current orbit to context stack)
9 T: (ORBIT) VAR>C T;
10 HEAD: SYS>C ( ( -- i ) \ move current system to context stack)
11 T: (SYSTEM) VAR>C T;
12 HEAD: ENC>C ( ( -- i ) \ move current encounter to ctxt stack )
13 T: (ENCOUNTER) VAR>C T;
14
15

```

20 73 1 IFIELD: ZHEADING (ship heading)
45 11 2 IFIELD: DEST-X (flux destination x coordinate)
45 17 2 IFIELD: DEST-Y (flux destination y coordinate)
61 11 1 IFIELD: ?HOME-PLANET (spacial context field)
61 27 1 IFIELD: ?ENCOUNTER-ON (encounter enable field)
23 17 1 IFIELD: ORBIT-MASK (system orbit mask)
32 1 1 AFIELD: P-SURF
32 4 1 AFIELD: P-LSEED
32 7 2 AFIELD: P-TSEED

2

5

```

0 ( SHPMOV - CONSTANTS 9/26/85) ( SHPMOV - @XYSHP !XYSHP !XHEADING @XHEADING ?XIDSEC.. 3-22-85)
1
2 3 C: RCOLLIDE ( collision radius)
3 0 C: PRTRADIUS ( radius of starport)
4 75 C: DEFRADIUS \ radius of sec map context - formerly 500
5 0 C: ORBRADIUS ( radius of orbit context)
6 75 C: ENCRADIUS ( radius of encounter context)
7 400 C: SYSRADIUS ( radius of system context)
8 64 C: VISLENGTH ( length of world window incl neb radii )
9 62 C: VISWIDTH ( width of world window " " " )
10 9 C: BVISOFF ( y world xlation to center origin mainview)
11 8 C: LVISOFF ( x world xlation to center origin mainview)
12 6 C: MAXSPEED ( maximum distance increment for ship movement)
13
14
15

```

HEAD: @XYSHP (-- x y \ get x and y loc of ship from object)
T: SHP>C INST-X @ INST-Y @ ICLOSE T;
head: !XYSHP (x y -- \ set x and y loc of ship in object)
t: SHP>C INST-Y ! INST-X ! ICLOSE t;
head: !XHEADING (heading -- \ save ship heading <0-8>)
t: SHP>C XHEADING C! ICLOSE t;
HEAD: @XHEADING (-- heading \ get ship heading)
T: SHP>C XHEADING C@ ICLOSE T;
HEAD: ?XIDSEC (-- t \ is the context sector map)
T: CONTEXT-ID# @ 3 = T;
HEAD: ?XIDENC (-- t \ is the context encounter)
T: CONTEXT-ID# @ 4 = T;

6

9

```

0 ( SHPMOV - ?* ?-? ?PLANET ?ENCOUNTER ?SHIP ?ORBIT ... 10/15/85) ( CONTEXT MODULE: PLANET AWK 10/22/85 )
1
2 HEAD: ?* ( -- t \ is the current object a star?) HEAD: ORB-DISP ( -- )
3 T: @INST-CLASS 24 = T; T: NULL !XSHIP XABS OFF YABS OFF INIT-ICONLIST
4 HEAD: ?-? T: ?* NOT T; ORB>C IOPEN +ICONBOX CDROP ICLOSE
5 HEAD: ?PLANET ( -- t \ is the current object a planet?) -5000 DUP ANCHOR D! ORBRADIUS LOCRAADIUS !
6 T: @INST-CLASS 32 = T; XABS OFF YABS OFF T;
7 HEAD: ?ENCOUNTER ( -- t \ is the current object an encounter?)
8 T: @INST-CLASS 61 = T; HEAD: .AUXSHIP ( -- )
9 HEAD: ?SHIP ( -- t \ is the current object the starship?) T: CONTEXT-ID# @ 2 =
10 T: @INST-CLASS 20 = T; IF WHITE !COLOR (RADIUS) @ >R
11 HEAD: ?*ORPLAN ( -- t \ star or planet?) XABS @ 36 I * / -35 MAX 32 MIN 120 +
12 T: ?* ?PLANET OR T; YABS @ 36 R > * / -35 MAX 32 MIN 161 + LXPLOR
13 HEAD: ?HYPER ( -- f \ is the context hyperspace? ) THEN T;
14 T: CONTEXT-ID# @ 3 = T;
15

```

7

10

```

0 ( CONTEXT MODULE: PLANET AWK 10/22/85) Encounter Context ifields AWK 7/16/85 )
1
2 HEAD: SURF>CLRMAP ( -- N ) 25 11 1 IFIELD: VES-HEADING
3 T: PLN>C P-SURF C@ 61 21 1 IFIELD: #AT-A-TIME
4 4 OVER = IF 5 ELSE 1 OVER = IF 4 ELSE 3 OVER = IF 3 ELSE 61 22 1 IFIELD: ?VES-CLASS0 61 23 1 IFIELD: ?VES-CLASS1
5 P-LSEED C@ IF 1 ELSE 2 THEN THEN THEN THEN SWAP DROP ICLOSE T; 61 24 1 IFIELD: ?VES-CLASS2 61 25 1 IFIELD: ?VES-CLASS3
6 61 19 1 IFIELD: MAX#SHIPS 61 20 1 IFIELD: #SHIPS-DESTROYED
7 HEAD: ORBSETUP ( -- )
8 T: (COLLIDE) VAR>C CI (ORBIT) 1.5! 52 18 1 IFIELD: BASE-EDL
9 IOPEN 32 0 IFIND DROP CI (PLANET) 1.5! CDROP ICLOSE T;
10 17 23 3 IFIELD: NAV
11 HEAD: ORBCLEANUP ( -- ) 16 27 1 IFIELD: ^NAV
12 T: ?PORT OFF SKIP2NEST OFF
13 ORB>C INST-X @ XTRNS ! INST-Y @ YTRNS ! ICLOSE
14 NULL (PLANET) 1.5! NULL (ORBIT) 1.5! 2NEST OFF
15 SET-'EXTERNAL ! 'AUX-CLEAN @ EXECUTE T;

```

8

11

```

0 ( Context Module: Planet 10-07-85) ( Encounter Context GET-VESSEL MAKE-VESSELS AWK 7/11/85 )
1 CASE GET-VESSEL
2 HEAD: ORBITSETUP ( -- ) 0 IS ?VES-CLASS3 1 IS ?VES-CLASS2 2 IS ?VES-CLASS1
3 T: 6 ' NS-MESS MODULE OTHERS ?VES-CLASS0
4 PLN>C IOPEN 12 0 IFIND NOT ICLOSE
5 IF SURF>CLRMAP HEAD: VCOORDS ( -- x y \ tighter radius for vessel coords if )
6 @INST-SPECIES 0 18 WITHIN ( encounter is orbital. )
7 IF CI DROP ELSE P-TSEED @ THEN ( #clrmap gseed --) T: ENCRADIUS (PLANET) 1.5@ OR
8 DUP GLOBALSEED ! IF 5 / THEN DUP NEGATE SWAP 1+ 2DUP RRND >R RRND R> T;
9 ' SET-PLANET MODULE
10 ' INIT-ORBIT MODULE HEAD: MAKE-VESSELS ( ( n -- n ) )
11 ' MROTATE 'EXTERNAL-EVENTS ! PLHI OFF T: #AT-A-TIME C@ DUP 2- SWAP 1+ RRND 1 MAX 0 DO 25
12 THEN BEGIN 0 4 RRND GET-VESSEL C@ ?DUP UNTIL
13 ICLOSE 6 ' NF-MESS MODULE T; IOPEN 1 *CREATE VCOORDS INST-Y ! INST-X !
14 0 7 RRND VES-HEADING C! ICLOSE
15 LOOP T;

```


12

```

0 ( Encounter Context KILL-VESSELS
1
2 HEAD: KILL-VESSELS ( { n -- n } )
3 T: ?CHILD
4 IF IOPEN
5   BEGIN @INST-CLASS 25 =
6     IF IDELETE 0 ELSE INEXT ?FIRST THEN ?NULL OR
7     UNTIL ICLOSE
8 THEN T;
9
10
11
12
13
14
15

```

13

```

0 ( Context Module: Aux Graphics P-SURF>COLOR
1
2 CASE P-SURF>COLOR
3   1 IS VIOLET 2 IS BLUE 3 IS WHITE 4 IS ORANGE
4 OTHERS BROWN
5
6 HEAD: >CONS-DISP ( -- )
7 T: CONTEXT-ID# @ 4 =
8   IF ' CSCALE MODULE
9   ELSE >ICONFONT
10     8 48 YWLD:YPIX D! 8 32 XWLD:XPIX D!
11     XLLDEST OFF YLLDEST OFF
12 THEN T;
13
14
15

```

14

```

0 ( Encounter Context .A-VESS .AUXENC
1 EXIT
2 HEAD: .A-VESS ( { i -- i } )
3 T: INST-X @ 36 ENCRADIUS */ 120 +
4   INST-Y @ 36 ENCRADIUS */ 161 + LPLOT T;
5
6 HEAD: .AUXENC ( { i -- i } )
7 T: BLACK ERASE-AUXILLARY
8 (PLANET) 1.5@ 2DUP D0= NOT
9 IF >C+S P-SURF C@ ICLOSE P-SURF>COLOR !COLOR
10 119 161 3 FILLCIRC 120 161 3 FILLCIRC
11 ELSE 2DROP THEN
12 IFIRST LT-GREEN !COLOR
13 BEGIN @INST-CLASS 25 =
14 IF .A-VESS THEN INEXT ?FIRST
15 UNTIL 6 #AUX ! T;

```

15

```

AWK 10/17/85) ( ENCOUNTER CONTEXT ENCSETUP ENCCLEANUP AWK 11/04/85)

HEAD: ENCSETUP ( -- )
T: (COLLIDE) 1.5@ (ENCOUNTER) 1.5! (SYSTEM) 1.5@ D0=
IF ' NOP 'AUX-CLEAN ! THEN
ENC>C @INST-SPECIES ORGIADDR 1.5@ ?HOME-PLANET C@ 1+
' OVINIT-COMM MODULE
MAKE-VESSELS ( .AUXENC ) ICLOSE T;

HEAD: ENCCLEANUP ( -- )
T: ENC>C INST-X @ XTRNS ! INST-Y @ YTRNS ! (AORIGINATOR) VAR>C
BASE-EDL C@ EDL @ 2DUP < 5 * >R > -5 * R> + ( SET NEW BEDL IF)
0 MAX 100 MIN BASE-EDL C! ICLOSE ( DELTA EDL < > 0 )
KILL-VESSELS ICLOSE NULL (ENCOUNTER) 1.5!
' NOP 'EXTERNAL-EVENTS ! 'AUX-CLEAN @ EXECUTE #AUX ON
NULL (AORIGINATOR) 1.5! >CONS-DISP T;

```

16

```

AWK 11/04/85) ( Encounter Context ENC-DISP AWK 10/17/85)

HEAD: ENC-DISP ( -- )
T: XABS OFF YABS OFF NULL !XYSHIP INIT-ICONLIST
ENC>C IOPEN +ICONBOX CDROP ICLOSE
-5000 DUP ANCHOR D! ENCRADIUS LOCADIUS ! ( .AUXSHIP ) T;

```

17

```

AWK 10/17/85) ( Starport Context AWK 10/11/85)

HEAD: PRSETUP ( -- )
T: TIME-PASSING OFF ?PORT ON T;

HEAD: PRTCLEANUP ( -- )
T: TIME-PASSING ON CFIGARRAYS
PLN>C INST-X @ XTRNS ! INST-Y @ YTRNS ! ICLOSE T;

```

18

```

0 ( >CONTEXT - MAKE-STAR KILL-STAR
1
2 HEAD: MAKE-STAR ( -- \ build a star object inside the current )
3   ( system)
4 T: SYS>C
5   INST-QTY @           ( flare-date)
6   24 @INST-SPECIES IOPEN 1 *CREATE
7   INST-QTY !
8   INST-X OFF INST-Y OFF ( position star to origin)
9   ICLOSE ICLOSE T;
10
11 HEAD: KILL-STAR ( -- \ delete the star object in the current )
12   ( system)
13 T: SYS>C IOPEN 24 0 IFIND IF IDELETE THEN ICLOSE ICLOSE T;
14
15

```

19

```

0 ( >CONTEXT - GET-ORBIT#
1
2 V: PREV-ORBIT V: ORBITS
3
4 HEAD: GET-ORBIT# ( -- n )
5 T: 9 PREV-ORBIT @ 1+ DO I BIT ORBITS @ AND
6   IF I PREV-ORBIT ! LEAVE THEN
7   LOOP PREV-ORBIT @ T;
8
9
10
11
12
13
14
15

```

20

```

0 ( >CONTEXT - >ORBIT
1
2 HEAD: (>ORBIT) ( ( i -- orb ) \ insert the current object into)
3   ( an orbit object. )
4 T: @INST-CLASS 32 =
5   IF GET-ORBIT# ELSE 0 THEN
6   11 4 ICREATE IEXTRACT 2OVER      ( box i box -- )
7   IINSERT                          ( box -- )
8   ?NULL IF CI' IINSERT IFIRST
9   ELSE CI' IINSERT
10  ?FIRST IF IPREV MAKEIST
11  ELSE IPREV
12  THEN
13  THEN INST-QTY ! T;              \ store orbit# or 0
14
15

```

21

```

6/13/85) ( >CONTEXT - ORBIT>                                     1/7/85)
HEAD: ORBIT> ( ( sys orb -- sys i ) \ delete the orbit object )
( and move it's content's to the system level. Pack )
( orbital encounters. Maintains order of items within orbit.)
T: IOPEN
  BEGIN
  ?-NULL
  WHILE
  IEXTRACT CI' <INSERT
  ICLOSE ?FIRST IF IPREV MAKEIST INEXT THEN IOPEN
  REPEAT
  ICLOSE IDELETE T;

HEAD: >ORBITS ( -- \ INSERT ORBIT OBJECTS INTO STARSYSTEM)
T: PREV-ORBIT OFF SYS>C ORBIT-MASK C@ ORBITS !
  IOPEN ' ?*ORPLAN ' (>ORBIT) EACH ICLOSE ICLOSE T;

```

22

```

6/17/85) ( >CONTEXT - >ORBITS ORBITS> SYSXY                      10/15/85)
HEAD: ?-ORBIT ( ( n -- n ) f )
T: @INST-CLASS 11 = @INST-SPECIES 4 = AND
  IF ORBIT> ELSE INEXT THEN T;

HEAD: ORBITS> ( -- \ DELETE ORBIT OBJECTS)
T: SYS>C IOPEN
  BEGIN ?LAST NOT
  WHILE ?-ORBIT
  REPEAT ?-NULL
  IF ?-ORBIT THEN CDROP ICLOSE T;

HEAD: SYSXY ( -- X Y \ GET SYSTEM X & Y FOR CURRENT STAR SYSTEM)
T: SYS>C INST-X @ INST-Y @ ICLOSE T;

```

23

```

6/14/85) ( >CONTEXT - SETRNDXY SETSYSXY                          10/15/85)
HEAD: SETRNDXY ( ( I -- I ) \ SET RANDOM X & Y LOCATION FOR )
( CURRENT OBJECT)
T: SYSRADIUS 5 - DUP NEGATE SWAP 2DUP RRND INST-X !
  RRND INST-Y ! T;

HEAD: SETELLIPSEXY ( ( I -- I ) )
T: INST-QTY @ 4 * 2+ >R I NEGATE
  I 1+ RRND DUP SYSRADIUS 36 #/ INST-X !
  I R> * SWAP DUP * - 0 SQRT
  0 2 RRND 2* 1- * SYSRADIUS 36 #/ INST-Y ! T;

HEAD: OCOORDS ( ( I -- I ) )
T: IOPEN ?PLANET ICLOSE
  IF SETELLIPSEXY ELSE INST-X OFF INST-Y OFF THEN T;

```

24

27

[illegible]

25

28

```

0 ( Context Module: Aux Graphics S-CLASS>COLOR      AWK 7/22/85 ) ( >CONTEXT - SYSSETUP  SYS-DISP  SYSCLEANUP      10/17/85)
1
2 CASE S-CLASS>COLOR                                HEAD: SYSSETUP ( -- )
3   ASCII M IS RED   ASCII K IS ORANGE  ASCII G IS YELLOW  T: (COLLIDE) 1.5@ (SYSTEM) 1.5! ' .AUXSYS 'AUX-CLEAN !
4   ASCII F IS WHITE ASCII A IS GREEN   ASCII B IS LT-BLUE  MAKE-STAR >ORBITS SETPLNXY .AUXSYS T;
5 OTHERS BLUE
6
7 HEAD: SYS-DISP ( -- )
8 T: INIT-ICONLIST
9   (SYSTEM) VAR>C IOPEN +ICONBOX CDROP ICLOSE
10  -5000 DUP ANCHOR D! SYSRADIUS LOCRAIDUS ! .AUXSHIP T;
11
12 HEAD: SYSCLEANUP ( -- )
13 T: SYS>C INST-X @ XTRNS ! INST-Y @ YTRNS ! ICLOSE
14 ORBITS> KILL-STAR NULL (SYSTEM) 1.5! #AUX ON T;
15

```

26

29

```

0 ( CONTEXT MODULE: AUX GRAPHICS .SUN .PLANET      AWK 10/15/85 ) ( >CONTEXT -      11/04/85)
1 HEAD: .SUN ( n -- )
2 T: S-CLASS>COLOR !COLOR 120 161 2 FILLCIRC T;
3
4 HEAD: .PLANET ( { i -- i } )
5 T: BEGIN
6   @INST-CLASS 11 =      ( is object a )
7   IOPEN ?PLANET AND DUP  ( planet box? ) HEAD: ?CAN-NAVIGATE ( -- f )
8   IF P-SURF C@ P-SURF>COLOR !COLOR THEN  ( get surf color ) T: TIME D@ NAV-TIME D@ D> T;
9   ICLOSE DUP NOT
10  IF INEXT ?FIRST OR      ( go on to next )
11    ELSE INST-X @ 36 SYSRADIUS */ 120 +  ( or center x,y )
12      INST-Y @ 36 SYSRADIUS */ 161 +  ( on aux screen )
13      1 1 1 FILLELLIP      ( and plot planet )
14  INEXT THEN      ( set up for next pass )
15  UNTIL T;

```

30

```

0 ( SHPMOV - >[CONTEXT] CTX>C ! [CONTEXT]
1
2 HEAD: >[CONTEXT] ( -- \ set context stack to frame pointer)
3 T: [CONTEXT] @ CXSP ! SET-CURRENT T;
4
5 HEAD: CTX>C ( (--i) \ copy current ctxt iaddr to top of stack )
6 T: (SHIPBOX) VAR>C T;
7
8 HEAD: ! [CONTEXT] ( -- \ set context stack frame pointer to tos)
9 T: CXSP @ [CONTEXT] ! T;
10
11
12
13
14
15

```

31

```

0 ( SHPMOV - XID-PLAN XID-ORB/*SEC ... ***misc*** 3-20-85)
1
2 HEAD: XIDPLN ( species -- 0 \ planet context id)
3 T: DROP 0 T;
4 HEAD: XIDORB/SEC ( species -- 1 or 3 \ orbit or sec context id)
5 T: 4 = IF 1 ELSE 3 THEN T;
6 HEAD: XIDSYS T: DROP 2 T;
7 HEAD: XIDENC T: DROP 4 T;
8 HEAD: XIDPRT T: DROP 5 T;
9
10
11
12
13
14
15

```

32

```

0 ( SHPMOV - CONTEXT-CASES RADIUS-CASES ***misc*** 7/25/85)
1
2 CASE CONTEXT-CASES ( species class -- context-id#)
3 32 IS XIDPLN
4 11 IS XIDORB/SEC
5 23 IS XIDSYS
6 61 IS XIDENC
7 18 IS XIDPRT ( actually docking into ship-config )
8 OTHERS UNRAVEL
9
10 CASE RADIUS-CASES ( context-id# -- radius )
11 1 IS ORBRADIUS
12 2 IS SYSRADIUS
13 4 IS ENCRADIUS
14 5 IS PRTRADIUS
15 OTHERS DEFRADIUS

```

33

```

5/22/85) ( SHPMOV - IVARCASES #CPARMS !CPARMS ***misc*** 5/20/85)
CASE IVARCASES ( xid# -- varaddr \ find address of iaddr )
( holding variable given context id#)
0 IS (PLANET)
1 IS (ORBIT)
2 IS (SYSTEM)
4 IS (ENCOUNTER)
OTHERS (SHIPBOX)

HEAD: #CPARMS ( i -- i ivar xid# radius \ compute context parms)
T: 2DUP >C+S @INST-SPECIES @INST-CLASS
CONTEXT-CASES DUP RADIUS-CASES >R >R
I IVARCASES R> R> ICLOSE T;
HEAD: !CPARMS ( i ivar xid# radius -- \ store context parms)
T: (RADIUS) ! CONTEXT-ID# ! >R 2DUP R> 1.5! (SHIPBOX) 1.5! T;

```

34

```

0 ( SHPMOV - .COORDINATES -ENDURIUM 10/11/85)
HEAD: .COORDINATES ( -- \ display ship starmap coordinates)
T: >DISPLAY XORMODE OFF >1FONT
3 GREY1 198 22 2DUP SWAP POS. POS.PXT ( erase old coords)
3 GREY1 198 48 POS.PXT
WHITE !COLOR ?CAN-NAVIGATE
IF @XYSHP ELSE 0 2000 RRND 0 2000 RRND THEN
*MAPSCALE / SWAP *MAPSCALE / 3 .R 48 xBLT ! . T;

HEAD: .E3 T: YELLOW !COLOR " FUEL SUPPLY LOW" .TTY BEEP T;
HEAD: .E2 T: ORANGE !COLOR " FUEL SUPPLY CRITICAL" .TTY BEEP T;
HEAD: .E1 T: PINK !COLOR " OUT OF FUEL" .TTY BEEP T;
CASE .EX 1 IS .E1 2 IS .E2 3 IS .E3 OTHERS NOP
HEAD: -ENDURIUM ( -- )
T: -1. 10*END D@ D+ 2DUP NULL DMAX 10*END D! 100 U/MOD 1+
SWAP 0= * .EX T;

```

35

```

0 ( SHPMOV - CONTEXT-CASES RADIUS-CASES ***misc*** 7/25/85) ( Space Dust SDUST >DUST ANK 10/23/85)
1
2 CASE CONTEXT-CASES ( species class -- context-id#)
3 32 IS XIDPLN
4 11 IS XIDORB/SEC
5 23 IS XIDSYS
6 61 IS XIDENC
7 18 IS XIDPRT ( actually docking into ship-config )
8 OTHERS UNRAVEL
9
10 CASE RADIUS-CASES ( context-id# -- radius )
11 1 IS ORBRADIUS
12 2 IS SYSRADIUS
13 4 IS ENCRADIUS
14 5 IS PRTRADIUS
15 OTHERS DEFRADIUS

72 C: XBND 120 C: YBND
255 C: XLIM 255 C: YLIM

V: SDUST -2 DP +!
130 C, 40 C, 56 C, 125 C, 215 C, 51 C,
212 C, 118 C, 178 C, 67 C, 200 C, 82 C,
58 C, 112 C, 134 C, 83 C, 159 C, 123 C,
61 C, 205 C, 127 C, 22 C, 39 C, 118 C,

HEAD: >DUST ( -- )
T: SDUST 24 + SDUST D0
0 XLIM RRND I C! 0 YLIM RRND I 1+ C!
2 +LOOP T;

```


36

39

```

0 ( Space Dust          ?DUST [.DUST]          AWK 9/23/85) ( SHPMOV - >NEWHEADXY          9/30/85)
1
2 HEAD: ?DUST ( x y -- x' y' )                HEAD: (>NEWHEADXY) ( heading x y -- )
3 T: YLIM MOD DUP 0 YBND WITHIN * SWAP          T: .AUXSHIP
4   XLIM MOD DUP 0 XBND WITHIN * SWAP T;        2DUP YABS ! XABS ! 2DUP !XYSHP
5                                                2DUP ILOCAL @ 1- POINT>ICON !IY !IX ?XIDENC NOT
6 HEAD: (.DUST) ( x y -- )                    IF 2DUP BVISOFF - SWAP LVISOFF - SWAP
7 T: XABS @ 500 + XWLD:XPIX D@ SWAP */          2SWAP VISLENGTH + SWAP VISWIDTH + SWAP !VISWINDOW
8   YABS @ 500 + YWLD:YPIX D@ SWAP */          ELSE 2DROP THEN ?HYPER
9   ROT + >R + R> ?DUST 2DUP *                IF .COORDINATES THEN
10  IF YBND - NEGATE SWAP XBND - NEGATE SWAP LPL0T  DUP HEADING ! 27 + !ID ( icon based on heading)
11  ELSE 2DROP THEN T;                          .AUXSHIP .MVS T;
12
13                                                HEAD: >NEWHEADXY ( heading x y -- \ move ship to new location)
14                                                ( and heading)
15 T: ?CHANGEXY IF (>NEWHEADXY) THEN T;

```

37

40

```

0 ( SHPMOV - .DUST      .MVS          11/04/85) ( SHPMOV - #SLOPE #RELXY          3-20-85)
1
2 HEAD: .DUST ( -- )                          HEAD: #SLOPE ( x y -- s 1 OR 0 \ compute the slope * 100 from )
3 T: GREY1 !COLOR                             ( origin to x,y and leave a code for undefined.)
4   SDUST @ + context-id# @ 2 = 16 * + SDUST DO  T: SWAP ( dy dx --)
5   I C@ I 1+ C@ (.DUST)                      ?DUP IF 100 SWAP */ 1
6   2 +LOOP T;                                ELSE DROP 0
7                                                THEN T;
8 : .MVS ( -- )
9 ?XIDENC NOT
10 IF >MAINVIEW VCLIPSET DARK
11   CONTEXT-ID# @ 2 4 WITHIN \ hyperspace or system
12   IF .DUST THEN
13     .LOCAL-ICONS V>DISPLAY
14     >DISPLAY DCLIPSET
15 THEN ;

```

38

41

```

0 ( SHPMOV - ?CHANGEXY .FLUX-ICON          7/17/85) ( SHPMOV - #QUAD          3-20-85)
1
2 HEAD: .FLUX-ICON ( -- )                    2 BASE !
3 T: @CRS @IX @IY WLD>SCR GREY1 !COLOR        HEAD: #QUAD ( x y -- q \ compute trig quadrant given x y)
4 CURSORSPEACE ABLT ! 1 WBLT ! 1 LBLT ! XORMODE OFF  T: 0< NOT IF 0011 ELSE 1100 THEN SWAP
5 NAV-SKILL @ 16 250 */ ?HAS-RING-DEVICE @ MAX 0 DO  0< NOT IF 1001 ELSE 0110 THEN AND
6   2DUP 15 RND CPT ROT + YBLT ! + XBLT ! BLT      DUP 1000 AND IF 2/
7   LOOP 2DROP !CRS T;                            ELSE DUP 0100 AND IF 1- THEN
8                                                    THEN T;
9 HEAD: ?CHANGEXY ( HEADING X Y -- [HEADING X Y 1] OR 0 \ IS NEW) DECIMAL
10 ( LOCATION AND HEADING DIFFERENT THAN CURRENT?)
11 T: >R >R >R
12 I HEADING @ =
13 I' XABS @ = AND
14 J YABS @ = AND NOT >V R> R> R> V>
15 ?DUP 0= IF 2DROP DROP 0 THEN T;

```


42

```

0 ( SHPMOV - #HEADING
1
2 HEAD: #HEADING ( Xo Yo Xd Yd -- h \ compute heading number )
3 ( <0-8> given location and destination. 8 is null heading. )
4 T: #RELY 2DUP D0= NOT
5 IF 2DUP #QUAD >R ( x y -- q \ Heading codes are: )
6 I 1 AND NOT IF SWAP THEN ( 3 2 1 )
7 ABS SWAP ABS SWAP ( norm to 1st quad \ \ ; / )
8 #SLOPE ( 4 - 8 - 0 )
9 IF DUP 41 < IF DROP 0 ELSE ( / ; / )
10 241 > IF 2 ELSE 1 THEN THEN ( 5 6 7 )
11 ELSE 2 ( undefined slope)
12 THEN
13 R> 1- 2* +
14 ELSE 2DROP 8 THEN T;
15

```

43

```

0 ( SHPMOV - HEADDXDY #HEADDXDY #NESTXY
1
2 V: HEADDXDY -2 DP +!
3 1 C, 0 C, 1 C, 1 C, 0 C, 1 C, -1 C, 1 C, -1 C, 0 C,
4 -1 C, -1 C, 0 C, -1 C, 1 C, -1 C,
5
6 HEAD: #HEADDXDY ( heading -- dx dy \ compute "unit" vector )
7 ( given heading)
8 T: 2* HEADDXDY + DUP +-@ SWAP 1+ +-@ T;
9
10 HEAD: #NESTXY ( radius heading -- x y \ compute entry coords)
11 ( given entry heading and context radius)
12 T: #HEADDXDY >R >R
13 1- R> OVER * NEGATE SWAP R> * NEGATE T;
14
15

```

44

```

0 ( SHPMOV - #VECTXY MVCLIPSET #DIST
1 HEAD: #VECTXY ( x y heading dist. -- x'y' \ compute new xy )
2 ( given location, heading and distance)
3 T: >R #HEADDXDY
4 I * SWAP R> * SWAP ( x y dx dy --)
5 ROT + >R + R> T;
6 HEAD: ACLIPSET ( -- \ set clipping window to auxill in mv buf)
7 T: 71 71 0 0 SETCLIPWINDOW T;
8
9 HEAD: MVCLIPSET ( -- \ set clipping window to mainview in full)
10 ( screen coordinates)
11 T: 191 75 72 4 SETCLIPWINDOW T;
12
13 HEAD: #DIST ( Xo Yo X1 Y1 -- n \ compute distance)
14 T: ROT - DUP * >R - DUP * R> + 0 SQRT
15 10*END D0 D0= NOT * T;

```

45

```

3-20-85) ( SHPMOV - #NEWHEADXY #SPEED #BOUNDXY 3-20-85)
HEAD: #SPEED ( -- n \ compute distance moved per iteration)
( based on location of ship relative to cursor)
T: XABS @ YABS @ XCRS @ YCRS @ #DIST MAXSPEED MIN T;
HEAD: #NEWHEADXY ( -- heading x y \ compute new heading & xy)
T: XABS @ YABS @ 2DUP XCRS @ YCRS @ #HEADING ( x y h --)
DUP 8 = IF DROP HEADING @ THEN
>R I #SPEED #VECTXY R> ( x y h --)
ROT ROT T;

```

46

```

3-20-85) ( SHPMOV - NEST-EFFECTS 3-19-85)

```

47

```

10/09/85) ( SHPMOV - UNNEST-EFFECTS support 6/13/85)

```

48

51

0 (SHPMOV - NEST-EFFECTS

3-19-85) (SHPMOV - NEST-FINIS-MESS

9/11/85)

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

49

52

0 (SHPMOV - UNNEST-FINIS-MESS

6/28/85) (SHPMOV - NEST-FINIS-MESS

9/11/85)

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

50

53

0 (SHPMOV - UNNEST-START-MESS

6/28/85) (SHPMOV - SEC-DISP DISP-SETUP

9/02/85)

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

HEAD: SEC-DISP (--)
T: IGLOBAL @ *GLOBAL @ = NOT
IF *SECS >C+S IOPEN INIT-ICONLIST
+ICONBOX IGLOBAL @ *GLOBAL ! CDROP ICLOSE
THEN -5000 DUP ANCHOR D! DEFRADIUS LOCADIUS ! T;

CASE DISP-SETUP
1 IS ORB-DISP
2 IS SYS-DISP
3 IS SEC-DISP
4 IS ENC-DISP
OTHERS NOP

54

```

0 ( SHPMOV - NEST-SETUP  NEST-SHIPXY
1 CASE NEST-SETUP ( xid# -- \ set up context)
2 \ 0 IS NOP ( bio tv, etc. )
3 1 IS ORBSETUP
4 2 IS SYSSETUP
5 \ 3 IS NOP \ no setup needed for hyperspace
6 4 IS ENCSETUP
7 5 IS PRTSETUP
8 OTHERS NOP
9
10 HEAD: NEST-SPACEXY ( n -- \ position ship within new space ctx )
11 T: 1 = NOT
12 IF HEADING @ (RADIUS) @ OVER #NESTXY 2DUP >R >R >NEWHEADXY
13 R> R> YCRS ! XCRS !
14 THEN T;
15

```

55

```

0 ( SHPMOV - NEST
1 : NEST ( i -- \ enter nested context i )
2 (SHIPBOX) VAR>C IOPEN CDROP
3 SHP>C IEXTRACT ( i s -- , ( i'i'--))
4 >[CONTEXT] 2SWAP >C+S ( s -- , ( i'i'--))
5 CI IINSERT ![[CONTEXT] ( -- , ( i'i'--))
6 CI #CPARMS !CPARMS CONTEXT-ID# @
7 DUP ' NS-MESS MODULE ( "starting transition..")
8 DUP NEST-SPACEXY ( position ship in ctxt)
9 DUP NEST-SETUP ( set up new context)
10 DUP DISP-SETUP ( display setup)
11 SET-'EXTERNAL ( external event )
12 >CONS-DISP ORGLIST >DUST HEADING @ XABS @ YABS @ (>NEWHEADXY)
13 ' NF-MESS MODULE ( "transition complete.")
14 XABS @ XCRS ! YABS @ YCRS ! 'KEY DROP
15 NULL (COLLIDE) 1.5! ;

```

56

```

0 ( SHPMOV - RAMVESSEL >ENCOUNTER DOCK ?BOX-NEST
1
2 HEAD: RAMVESSEL ( i -- \ handle vessel collision with ship)
3 T: >C+S CI @INST-SPECIES ICLOSE 24 = \ is debris?
4 IF 2DROP ( <-- ?GET-DEBRIS ) ELSE 2DROP THEN T;
5
6 HEAD: >ENCOUNTER ( i -- \ nest into active encounter or nop)
7 T: >C+S CI ?ENCOUNTER-ON C@ ICLOSE IF NEST ELSE 2DROP THEN T;
8
9 HEAD: ?BOX-NEST ( i -- )
10 T: >C+S IOPEN @INST-CLASS 24 = NOT ICLOSE
11 IF CI ICLOSE NEST ELSE ICLOSE THEN T;
12
13
14
15

```

57

```

10/10/85) ( SHPMOV - >FLUX-NODE 7/19/85)
HEAD: SET-FLUX-STATE ( x y ( i -- i ) \ set traversal flags )
T: IGLOBAL @ ?ICONS-AT ?DUP
IF 0 DO POINT>ICON @ID 255 =
IF 3 !IC @IL @IH >C+S 1 !INST-SPECIES ICLOSE THEN LOOP THEN T;

HEAD: >FLUX-NODE ( i -- \ move ship through flux)
T: >C+S
INST-X @ INST-Y @ SET-FLUX-STATE ( set source flux flags )
DEST-X @ DEST-Y @ SET-FLUX-STATE ( set target flux flags )
' OVFLUX-EFFECT MODULE ( flash effect to display )
HEADING @ DEST-X @ DEST-Y @ ( get destination )
ICLOSE 0 @ RRND RCOLLIDE 1+ ( arrive out of collis rad )
#VECTXY >NEWHEADXY ( move ship to new coords )
ORGLIST .MVS
TIME D@ 250 NAV-SKILL @ - 500 U* D+ NAV-TIME D! T;

```

58

```

10/22/85) ( SHPMOV - COLLIDE 9/24/85)
CASE COLLIDE ( i class -- \ handle class dependent collision)
11 IS ?BOX-NEST ( orbit)
18 IS NEST ( dock w/starport 18 is ship-config class)
23 IS NEST ( starsystem)
25 IS RAMVESSEL
45 IS >FLUX-NODE
61 IS >ENCOUNTER ( encounter)
OTHERS 2DROP ( nebula, planet)

```

59

```

10/17/85) ( SHPMOV - ?COLLISION 7/25/85)
HEAD: ?COLLISION ( -- i class 1 OR 0 \ check for collision)
T: @XYSHP RCOLLIDE ILOCAL @ ?ICONS-LOCUS ( nnnnn cnt -- )
?DUP IF NULL (COLLIDE) 1.5!
0 DO POINT>ICON
@IL @IH *SHIP D= ( not ship or )
@ID 51 131 WITHIN @IC DK-GREEN = AND OR NOT ( nebula )
CONTEXT-ID# @ 1 = NOT AND ( or in orbit )
IF @IL @IH (COLLIDE) 1.5! THEN
LOOP
(COLLIDE) 1.5@ OR
IF (COLLIDE) VAR>C CI @INST-CLASS
INST-X @ XTRNS ! INST-Y @ YTRNS ! ICLOSE 1
ELSE 0 THEN
ELSE 0 THEN T;

```

60

```

0 ( SHPMOV - ?DOUBLE-NEST                                AWK 10/17/85) ( SHPMOV - UNNEST-SHIPXY                                10/10/85)
1 HEAD: ?2NEST.ETC ( -- f )
2 T: (COLLIDE) 1.5@ D0=                                     ( a. if no collision ) HEAD: UNNEST-SPACEXY ( n -- \ position ship within prev ctxt)
3 2NEST @ 0= AND                                           ( b. and not double nested ) T: HEADING @ >R I
4 CONTEXT-ID# @ 1 = AND DUP T;                             ( c. and in orbital context ) XTRNS @ YTRNS @ R> RCOLLIDE 1+ #VECTXY >NEWHEADXY T;
5
6 HEAD: ?DOUBLE-NEST ( -- i class f \ force collide w/orbital enc)
7 T: ?2NEST.ETC SKIP2NEST @ XOR                             ( if not forcing skip2nest )
8 IF PLN>C IOPEN 61 0 IFIND DUP                             ( d. that has an encounter )
9 IF ?ENCOUNTER-ON C@ DUP                                   ( e. which is active )
10 IF CI (COLLIDE) 1.5!                                     ( f. make it the collidEE )
11 2NEST ON                                                  ( g. set double-nest flag )
12 THEN AND
13 THEN CDROP ICLOSE AND
14 THEN
15 IF (COLLIDE) VAR>C CI @INST-CLASS ICLOSE 1 ELSE 0 THEN T;

```

61

```

0 ( SHPMOV - ?STARPORT                                    AWK 9/11/85) ( SHPMOV - ?UNNEST ?DOUBLE-UNNEST                                10/17/85)
1
2 HEAD: ?STARPORT ( -- i class f \ force collide w/ starport ) HEAD: ?UNNEST ( -- t \ time to unnest?)
3 T: ?2NEST.ETC                                             T: ?HYPER
4 IF PLN>C IOPEN 12 0 IFIND DUP                             ( d. that has Starport then ) IF 0 ELSE ?ENCOUNTER
5 IF CI (COLLIDE) 1.5!                                     ( e. make Starport collidEE ) IF XABS @ YABS @ (RADIUS) @ ILOCAL @ ?ICONS-LOCUS
6 2NEST ON                                                  ( f. set double-nest flag ) DUP >R ?DUP
7 THEN CDROP ICLOSE AND                                     IF 0 DO DROP LOOP THEN R> 1 = ( no nearby alien ships? )
8 THEN                                                       ENC>C ?ENCOUNTER-ON C@ NOT OR ( encounter off? )
9 IF *SHIP-CONFIG 18 1                                     ( dock w/starort ship-config) MAX#SHIPS C@ #SHIPS-DESTROYED C@ OVER = ( all destroyed?)
10 58 FILE# ! 1 RECORD# !                                   SWAP 255 < AND OR ICLOSE DUP SKIP2NEST ! ( ltd amount? )
11 " LAUNCH" 18TN                                           ELSE XABS @ ABS YABS @ ABS MAX (RADIUS) @ >
12 SWAP CMOVE SET-CURRENT                                   THEN THEN T;
13 ELSE 0 THEN T;
14
15 HEAD: ?DOUBLE-UNNEST ( -- f \ unnest twice from orbital enc? )
T: 2NEST @ SKIP2NEST @ XOR CONTEXT-ID# @ 1 = AND T;

```

62

```

0 ( SHPMOV - UNNEST-CLEANUP                                7/19/85) ( SHPMOV - UNNEST                                10/21/85)
1
2 CASE UNNEST-CLEANUP ( xid# -- \ set up context)
3 \ 0 IS NOP ( PLNCLEANUP ) \ handled by Bob's code
4 1 IS ORBCLEANUP
5 2 IS SYSCLEANUP
6 \ 3 IS NOP \ because we never leave space!
7 4 IS ENCCLEANUP
8 5 IS PRTCLEANUP
9 OTHERS NOP
10
11
12
13
14
15

```

```

: UNNEST ( -- \ leave nested context to next level up)
(SHIPBOX) VAR>C IOPEN CDROP
SHP>C IEXTRACT ICLOSE ( s -- , { i'i"--})
CONTEXT-ID# @ >R
I ' US-MESS MODULE ( "leaving ____")
I UNNEST-CLEANUP
>[CONTEXT] ICLOSE ( s -- , { i'i"--} \ pop context)
CI IINSERT ![CONTEXT] ( -- , { i'i"--} \ insert ship)
CI #CPARMS !CPARMS ( set context parms)
CONTEXT-ID# @ UNNEST-SPACEXY ( reposition ship)
SET-'EXTERNAL ( external event )
DISP-SETUP ( display setup)
ORGLIST >DUST HEADING @ XABS @ YABS @ (>NEWHEADXY)
R> ' UF-MESS MODULE ( "unnest complete")
XABS @ XCRS ! YABS @ YCRS ! 'KEY DROP ;

```

65

72

```

0 ( SHPMOV INITIALIZERS                                AWK 10/09/85)
1
2 XABS @ XCRS ! YABS @ YCRS !
3 ' .FLUX-ICON ' .FLUX-ICON !
4 CCLR *SECS >C ![[CONTEXT]
5 0 !%HEADING HEADING OFF 3 CONTEXT-ID# !
6 XLLDEST OFF YLLDEST OFF
7 NULL (COLLIDE) 1.5!
8
9
10
11
12
13
14
15

```

73

```

0 ( SHPMOV ----- OVERLAY SUFFIX -----10/23/85)
1 DISPOSE CLOSE-OVERLAY
2 106 OVERLAY SHPMOV-OV
3 SHPMOV-OV FORTH DEFINITIONS
4
5 : MANEUVER ( -- \ do maneuver tasks)
6 SHPMOV-OV SHPMOV-VOC CHK-MOV
7 IF (GET-CX-AUX)
8 ' CLEANUP-MOV ' SET-DESTINATION ' SETUP-MOV DOTASKS
9 THEN FTRIG OFF ;
10 : SETUP-MOV ( -- ) SHPMOV-OV SHPMOV-VOC SETUP-MOV ;
11 : GET-HAUX SHPMOV-OV SHPMOV-VOC (GET-AUX) ;
12 : ov.MVS ( -- ) SHPMOV-OV SHPMOV-VOC .MVS ;
13 : S>PORT ( -- ) SHPMOV-OV SHPMOV-VOC COME-HITHER ;
14 \ S>PORT
15 OV-CANCEL

```