

MYWS	EQU	CPURAM	
FASTWS	EQU	CPURAM	
STOPFL	EQU	CPURAM+>20	0=STOP ?=NO STOPPING
TGCOOL	EQU	CPURAM+>21	TEMP. GAGE COOLING TIMER
JOYFLG	EQU	CPURAM+>22	SET IF WE'VE USED JOYSTICK
CHEATF	EQU	CPURAM+>23	CHEAT FLAG IF WE'VE COME FROM CHEAT MODE
M1DANG	EQU	CPURAM+>24	MONSTER 1 DANGEROUS 0=NO
M2DANG	EQU	CPURAM+>25	MONSTER 2 DANGEROUS 0=NO
GPCSAV	EQU	CPURAM+>26	FOR RETURNING TO GROM CODE
M1MTIM	EQU	CPURAM+>28	MONSTER 1 CHANGE DIR TIME
M2MTIM	EQU	CPURAM+>29	MONSTER 2 CHANGE DIR TIME
SCROLD	EQU	CPURAM+>2A	SCROLL DELAY VALUE
SCROLC	EQU	CPURAM+>2B	SCROLL COUNTER
GALWAT	EQU	CPURAM+>2C	GALLONS OF WATER REMAINING
OBJFLG	EQU	CPURAM+>2D	OBJECT FLAG 0-NO OBJ 1-OBJECT
CANDLY	EQU	CPURAM+>2E	CREW ANIMATION DELAY
WGTIMV	EQU	CPURAM+>2F	WATER GAGE TIMER DELAY VALUE
LDNO	EQU	CPURAM+>30	LOAD NUMBER FOR WALL INCREMENTING
CMVDIR	EQU	CPURAM+>31	COMPUTER MOVE DIRECTION SAVED FOR DEMO
SHOTFL	EQU	CPURAM+>32	SHOT FLAG 1-ON, 0-OFF
PAIRFL	EQU	CPURAM+>33	PAIR FLAG 1-VERT, 2-SIDES
MN1FLG	EQU	CPURAM+>34	MONSTER 1 FLAG 0=NO MONSTER
M1CTIM	EQU	CPURAM+>35	MONSTER 1 TIMER BYTE
MN1PNT	EQU	CPURAM+>36	POINTER TO GROM FOR MONSTER PATTERN
TEMPRT	EQU	CPURAM+>38	EMERGENCY RETURN LINK SAVE
WHICHL	EQU	CPURAM+>3A	0-ENGLISH ?-NOT ENGLISH
SPADR	EQU	CPURAM+>3C	FOR INTERRUPT DRIVEN SPEECH
SPLEN	EQU	CPURAM+>3E	LENGTH OF SPEECH DATA BLOCK
MN2FLG	EQU	CPURAM+>40	MONSTER 2 FLAG
M2CTIM	EQU	CPURAM+>41	MONSTER 2 TIMER BYTE
MN2PNT	EQU	CPURAM+>42	MONSTER 2 GROM POINTER
TGAGPT	EQU	CPURAM+>44	TEMP. GAGE POINTER FOR SIT
WGAGPT	EQU	CPURAM+>46	H2O GAGE POINTER FOR SIT
CREWS	EQU	CPURAM+>48	MOON MINES REMAINING
GAMELV	EQU	CPURAM+>49	GAME LEVEL (0-59)
GPLRA	EQU	CPURAM+>4A	GPL RETURN ADDRESS
WGCHAR	EQU	CPURAM+>4C	CURRENT WATER GAGE CHARACTER
WGTIME	EQU	CPURAM+>4D	WATER GAGE DECREMENT TIMER
M1HITS	EQU	CPURAM+>4E	# HITS ON MONSTER 1
M2HITS	EQU	CPURAM+>4F	# HITS ON MONSTER 2
HITLIM	EQU	CPURAM+>50	NUMBER OF HITS ALLOWED ON MONS
MKILED	EQU	CPURAM+>51	TOTAL KILLED
LETMON	EQU	CPURAM+>52	0-NO MONSTERS 1-MONSTERS
OBSPCH	EQU	CPURAM+>53	0-ALLOW OBJECT SPEECH 1-DON'T ALLOW
*****			THE NEXT 8 BYTES ARE FLAGS FOR THE WATER EMERGING FROM THE TUNNEL
OBEENB	EQU	CPURAM+>57	OBJECT BEEN BEFORE FLAG 0=NOT BEEN
WONSCN	EQU	CPURAM+>58	WATER ON SCREEN FLAG 0-NOT ON
WBEENB	EQU	CPURAM+>59	WATER BEEN BEFORE FLAG 0-NOT BEEN BEFORE
WPIXLS	EQU	CPURAM+>5A	# OF PIXELS MOVED (0-5)
WSIZCH	EQU	CPURAM+>5B	# OF WATER SIZE CHANGES
WTLOSS	EQU	CPURAM+>5C	WATER LOSS FLAG 1-DO NOT DECREMENT
WATPNT	EQU	CPURAM+>5E	WATER GROM POINTER

CRWOUT	EQU	CPURAM+>60	CREW MEMBER OUT FLAG 0-NOT OUT
CRWDIR	EQU	CPURAM+>61	CREW DIR 0-STILL 1-LEFT 2-RIGHT
CPCNTR	EQU	CPURAM+>62	CREW PATTERN COUNTER 0, 1, OR 2
CPINC	EQU	CPURAM+>63	CREW PATTERN INCR. -1 OR 1
CGRMPT	EQU	CPURAM+>64	CREW GROM POINTER (2 BYTES)
CSHOTF	EQU	CPURAM+>66	CREW SHOT FLAG 0-NO SHOT FIRED
FLAG5	EQU	CPURAM+>67	5,000 POINTS FLAG 0-NEW CREW
CLSLIM	EQU	CPURAM+>68	CREW LASER SHOT LIMIT IN TUNNEL (12)
INTFLG	EQU	CPURAM+>69	INTERRUPT FLAG 0-NOT IN INTERRUPT

```

LANGPT EQU CPURAM+>6A      POINTER TO LANGUAGE CURRENTLY IN
LANG    EQU CPURAM+>6C      GROM FORMATTER FOR LANGUAGES POINTER
DEMOFL  EQU CPURAM+>6E      DEMO FLAG  0=NO DEMO  1=DEMO
CMVDLY  EQU CPURAM+>6F      COMPUTER MOVE DELAY VALUE
KEYBRD  EQU CPURAM+>74
KEY      EQU CPURAM+>75
JOYY    EQU CPURAM+>76
JOYX    EQU CPURAM+>77
RANDOM   EQU CPURAM+>78      TIMER BYTE
TIMER   EQU CPURAM+>79      FOR AUTO MOTION
MOTION  EQU CPURAM+>7A      STATUS REGISTER
STATUS  EQU CPURAM+>7C      INTERRUPT WORKSPACE
INTWS   EQU CPURAM+>C0      BUILT IN SEED
SEED    EQU CPURAM+>C0
DISINT  EQU CPURAM+>C2      DISABLE SPRITE MOTION AND QUIT FUNCTION
EXINTR  EQU CPURAM+>C4      EXTERNAL INTERRUPT ROUTINE WINDOW
SVVDP1  EQU CPURAM+>D4      TO SAVE VDP R1 IF KEY PRESSED
SCNTIM  EQU CPURAM+>D6      SCREEN TIME OUT COUNTER
GPLWS   EQU CPURAM+>E0      ADDRESS OF GPL WORKSPACE
PAGE

```

```

*****
* OTHER EQUATES *
*****

```

```

PTBL2 EQU >E000      PARAMETER PASSING TABLE 2
GCHRLD EQU PTBL2+>4  GROM ADDRESS OF LOAD DATA
CWLFT1 EQU >6F06      CREW MEMBERS LEFT DATA
CWLFT2 EQU CWLFT1+60
BLANKM EQU CWLFT2+60  BLANK BORDER CHARACTERS
*----- MONSTER PATTERNS START HERE -----*
PTBL1 EQU >8000      PARAMETER PASSING TABLE 1
ZYGD   EQU PTBL1+>20  LEGENDARY ZYGONAUGHT PATTERN
ZYGOEY EQU ZYGD+320
CREWU  EQU ZYGOEY+320  CREW FIRING POSITION
CREWD  EQU CREWU+96
CREWR  EQU CREWD+96    START OF CREW RIGHT ANIMATION
CREWL  EQU CREWR+96    START OF CREW LEFT ANIMATION
OBJECS EQU CREWL+96    START ADDRESS OF ALL OBJECTS IN GROM
WATER  EQU OBJECS      WATER CHARACTERS
TREAS  EQU WATER+256   TREASURES TO FOLLOW
SILVER EQU TREAS       SILVER CHARACTERS
DIAMON EQU SILVER+256  DIAMOND CHARS
AMETHY EQU DIAMON+256  AMETHYST CHARACTERS
MONS   EQU AMETHY+256  START ADDRESS OF ALL MONSTERS IN GROM
PAGE

```

```

*****
* REGISTER EQUATES *
*****

```

```

TEMPO EQU R0
VDPADD EQU R0          VDP ADDRESS
RCOUNT EQU R3          NUMBER OF BYTES
WCOUNT EQU R3          NUMBER OF BYTES
RLOC   EQU R4          CPU BUFFER
WLOC   EQU R4          CPU BUFFER
COUNT EQU R5          TO MOVE RCOUNT OR WCOUNT TO
DIGHB  EQU R6
TEMP   EQU R6          TO MOVE VDPADD TO
CARYHB EQU R7

```

```

*****
* WORKSPACE EQUATES *
*****

```

```

ROLB   EQU MYWS+1
VADDLB EQU ROLB

```

```

R1LB EQU MYWS+3
R2LB EQU MYWS+5
R3LB EQU MYWS+7
R4LB EQU MYWS+9
R5LB EQU MYWS+11
GR5LB EQU GPLWS+11
R6LB EQU MYWS+13
DIGTLB EQU R6LB FOR SCORE ROUTINE
R7LB EQU MYWS+15
GR7LB EQU GPLWS+15
CARYLB EQU R7LB FOR SCORE ROUTINE
R8LB EQU MYWS+17
GR8LB EQU GPLWS+17
R9LB EQU MYWS+19
RALB EQU MYWS+21
GRALB EQU GPLWS+21
RBLB EQU MYWS+23
RCLB EQU MYWS+25
RDLB EQU MYWS+27
RELB EQU MYWS+29
RFLB EQU MYWS+31
PAGE

```

```

*****
* SOUND TABLE EQUATES *
*****

```

```

FIRSND EQU >1000 FIRING SOUND
EXPSND EQU FIRSND+94 EXPLOSION SOUND
TNKSND EQU EXPSND+54 TANK MOTION SOUND
WLKSND EQU TNKSND+14 CREW RUNNING SOUND
SNDOFF EQU WLKSND+14 ALL SOUNDS OFF
ZYCSND EQU SNDOFF+6 ZYGONAUGHT COMING
ZYGSND EQU ZYCSND+41 ZYGONAUGHT GOING
HITSND EQU ZYGSND+41 MONSTER HIT SOUND
GAGSND EQU HITSND+13 WATER GAGE FULL SOUND
BONSND EQU GAGSND+39 BONUS POINTS SOUND
***** EQU BONSND+39

```

```

*****
* SPEECH PHRASES AVAILABLE IN MOONMINE *
*****

```

```

* ADVANCE TO NEXT LEVEL 1
* BONUS POINTS GAINED 2
* EXTRA CREW MEMBER 3
* GOOD SHOT, CAPTAIN 4
* MONSTER DESTROYED 5
* WAY TO GO, CAPTAIN 6
* LASERS OVERHEATED 7
* MONSTER DAMAGED SHIP 8
* MONSTER ATTACKED CREW 9
* CREW MEMBER LOST 10
* DETECTING UNKNOWN OBJECT 11
* ZYGONAUT APPROACHING 12
* CONTINUE GAME, CAPTAIN 13
* COOLANT IS RUNNING LOW 14
* WATER AHEAD 15
* WATER TANKS FULL 16
* OUT OF WATER, CAPTAIN 17
* CONGRATULATIONS, CAPTAIN 18
* HA HA HA HA 19
* YOU'LL NEVER GET ME 20

```

```

*****
* TEXT PHRASES AVAILABLE IN MOONMINE *
*****

```

*ADVANC	1	ADVANCE TO NEW LEVEL
*BONUSP	2	BONUS POINTS GAINED
*EXTRAS	3	EXTRA SHIP
*LASERS	4	LASERS OVERHEATED
*MONSTE	5	MONSTER DESTROYED SHIP!
*MONSTC	6	MONSTER ATTACKED CREW!
*CREWLO	7	CREW MEMBER LOST
*DETECT	8	DETECTING UNKNOWN OBJECT
*ZYGONA	9	ZYGONAUGHT APPROACHING
*CONTIN	10	CONTINUE GAME CAPTAIN
*COOLAN	11	COOLANT IS RUNNING LOW
*WATERA	12	WATER AHEAD
*WATERT	13	WATER TANKS FULL
*OUTOFW	14	OUT OF WATER, CAPTAIN
*CONGRA	15	CONGRATULATIONS, CAPTAIN

PAGE

 * B Y T E S T A T E M E N T S *

MBOUND	BYTE	48, 64, 64, 176	MONSTER BOUNDS 0
	BYTE	48, 64, 64, 176	MONSTER BOUNDS 1
	BYTE	44, 68, 60, 180	MONSTER BOUNDS 2
	BYTE	40, 72, 56, 184	MONSTER BOUNDS 3
	BYTE	36, 76, 52, 188	MONSTER BOUNDS 4
	BYTE	32, 80, 48, 192	MONSTER BOUNDS 5
	BYTE	28, 84, 44, 196	MONSTER BOUNDS 6
	BYTE	24, 88, 40, 200	MONSTER BOUNDS 7
	BYTE	16, 96, 32, 208	MONSTER BOUNDS 8

***** T H A T W A S 3 6 B Y T E S O F M O N S T E R B O U N D A R I E S

H20	BYTE	>20	FOR SPACE BAR HIT
H39	BYTE	>39	ASCII '9'
H60	BYTE	>60	FOR SPEECH ROUTINE
HA2	BYTE	>A2	ALLOW NO SCREEN OUTPUT
HAA	BYTE	>AA	
HAB	BYTE	>AB	
HAD	BYTE	>AD	
HCO	BYTE	>CO	FOR SPRITES OFF SCREEN
HE2	BYTE	>E2	ALLOW SCREEN OUTPUT
H10	BYTE	>10	DISABLE QUIT KEY
HFF	BYTE	>FF	NEGATIVE ONE

*
 MONTIM BYTE >30, >30, >28, >24, >20, >16, >14, >10, >72 MSTR TIME LIMS
 *

MONVEL	BYTE	10, -10, 15, -15	0 POSSIBLE MONSTER VELOCITIES
	BYTE	12, -12, 17, -17	1 POSSIBLE MONSTER VELOCITIES
	BYTE	14, -14, 19, -19	2 POSSIBLE MONSTER VELOCITIES
	BYTE	16, -16, 21, -21	3 POSSIBLE MONSTER VELOCITIES
	BYTE	18, -18, 23, -23	4 POSSIBLE MONSTER VELOCITIES
	BYTE	20, -20, 25, -25	5 POSSIBLE MONSTER VELOCITIES
	BYTE	22, -22, 27, -27	6 POSSIBLE MONSTER VELOCITIES
	BYTE	24, -24, 29, -29	7 POSSIBLE MONSTER VELOCITIES
	BYTE	26, -26, 31, -31	8 POSSIBLE MONSTER VELOCITIES
	BYTE	28, -28, 33, -33	9 POSSIBLE MONSTER VELOCITIES
	BYTE	30, -30, 35, -35	10 POSSIBLE MONSTER VELOCITIES
	BYTE	31, -31, 36, -36	11 POSSIBLE MONSTER VELOCITIES
	BYTE	32, -32, 37, -37	12 POSSIBLE MONSTER VELOCITIES
	BYTE	33, -33, 38, -38	13 POSSIBLE MONSTER VELOCITIES
	BYTE	34, -34, 39, -39	14 POSSIBLE MONSTER VELOCITIES
	BYTE	35, -35, 40, -40	15 POSSIBLE MONSTER VELOCITIES
	BYTE	37, -37, 42, -42	16 POSSIBLE MONSTER VELOCITIES
	BYTE	39, -39, 44, -44	17 POSSIBLE MONSTER VELOCITIES
	BYTE	41, -41, 46, -46	18 POSSIBLE MONSTER VELOCITIES

BYTE 43, -35, 55, -50 19 POSSIBLE MONSTER VELOCITIES

*
SCROLL BYTE 07, 07, 07, 06, 06, 06, 05, 05, 05, 04
BYTE 04, 04, 03, 03, 02, 02, 02, 02, 01, 01 SCROLL DELAY VALUES

*
ZSCORE BYTE >30, >30, >30, >30, >30, >30 ALL ZEROES
WGFULL BYTE >08, >08, >08, >08, >08, >08
BYTE >08, >08, >08, >08, >08 WATER GAGE FULL
TQEMPT BYTE >61, >61, >61, >61, >61
BYTE >61, >61, >61, >61, >61 TEMP. GAGE EMPTY
LVLCLR BYTE >04, >0C, >0D SCREEN LEVEL COLORS
MHITLM BYTE >01, >02, >03 MONSTER HIT LIMIT
WALLNC BYTE >1B, >1B, >1B, >1B, >1A, >1A NORMAL WALL COLORS
WALLFC BYTE >19, >19, >18, >18, >16, >16 FLASHED WALL COLORS
MNWDC BYTE >09, >06 MONSTER WOUNDED COLORS
TREASC BYTE >0F, >0C, >0D TREASURE COLORS
DEMOKY BYTE 2, 0, 13, 3, 5, 13, 1, 1 POSSIBLE KEYS PRESSED FOR DEMO
PAGE

* DATA STATEMENTS NEEDED *

MONTLX DATA 1, 4, 5, 5, 6, 8, 9, 10, 11 MONSTER X TOLERANCES
MONTHLY DATA 1, 6, 7, 8, 8, 10, 11, 12, 13 MONSTER Y TOLERANCES
MONPNT DATA >0100, >0100, >0090, >0080, >0070 MONSTER POINT VALUES
DATA >0060, >0050, >0040, >0030
CRUBND DATA >CCCE, >D4D6, >DCDE, >E4E6, >ECEE, >0000 CREW UP CHECK
CRDBND DATA >C6C4, >FOF1, >F2F3, >F4F5, >F600 CREW DN CHECK
CRLBND DATA >F3CE, >D6DE, >E6EE, >0000 CREW LT CHECK
CRRBND DATA >F4CC, >D4DC, >E4EC, >0000 CREW RT CHECK
CBINCV DATA >0804, >0808, >0800, >1008, >0808 CREW BOUNDARY INCREMENT VALUES
REDCNT DATA >6262, >6262, >6262, >6262, >6262, >6262, >6262 RED CENTER
BLKCNT DATA >Q404, >0404, >0404, >0404, >0404, >0404, >0404 BLK CENTER
H0000 DATA >0000 FULL WORD OF ZEROES
PAGE

* START OF MAINLINE ROM CODING *

EVEN
MOON LIM1 0
MOV R11, @GPLRA SAVE GPL RETURN ADDRESS FOR END
LWPI MYWS USE MY WORKSPACE AT >B300
BL @SAVGPL SAVE GPC AND CLEAR SPEECH
MOV @HA2, @SVVDP1 TO PRESERVE VDPR1
* MOV @H10, @DISINT DISABLE AUTO MOTION AND QUIT
LI R4, CHKMON ADDRESS OF ALL MY INTERRUPT ROUTINES
MOV R4, @EXINTR VECTOR FOR EXTERNAL INTERRUPT. RTN
LIM1 2
PAGE

* THIS CODE DOES THE INITS FOR THE START OF THE GAME *
* THIS MODULE IS EXECUTED ONLY WHEN COMING FROM GROM!! *

INITS MOV @H00, @LDNO START WITH LOAD 1
MOV @H01, @SCROLC SCROLL DELAY COUNTER
BL @DRWERS DRAW THE INITIAL WALLS
BL @CLRSCR ZERO OUT THE SCORE
MOV @CHEATF, @CHEATF DID WE COME FROM CHEAT MODE?
JEG INICON NO, SO CONTINUE
LI R1, >2A YES, SO WE NEED ASTERISK
BL @WRTVCL SIGNIFY NO START OF NEW GAME
DATA SIT+>23B, 1, R1LB

*

```

INICON  MOVB @H00,@FLAG5      HAVEN'T HIT 5,000 POINTS YET
        MOVB @CREWS,@CREWS    CREWS INITIALIZED IN CHEAT MODE?
        JNE GSHWTK            YES, SO DON'T INITIALIZE HERE
        MOVB @H05,@CREWS      NO, SO START WITH FIVE
GSHWTK  BL @SHWCRW            SHOW # OF REMAINING CREWS ON THE SCREEN
        MOVB @GAMELV,@GAMELV  GAME LEVEL INITIALIZED IN CHEAT?
        JNE GCLRLV           YES, SO DON'T INIT HERE
        MOVB @H00,@GAMELV     NO, SO START WITH LEVEL ZERO!
GCLRLV  SB @H01,@GAMELV      WILL BE INCREMENTED RIGHT AWAY
        BL @NEWLVL           PUT SCREEN LEVEL ON SCREEN
        MOVB @SCROLD,@SCROLC  SET THE DELAY COUNTER FOR SCROLL
        BL @CLRHI            ZERO OUT THE HIGH SCORE, TOO!
        BL @CLGAGE           SET THE WATER AND TEMP GAGES
        BL @SCRNON           TURN ON THE SCREEN
        MOVB @HOC,@MOTION     ALLOW SIX SPRITES IN MOTION
        B @MAINLP           GO TO MAIN PLAYING LOOP
        PAGE

```

```

*****
* THIS CODE DOES THE INITIS FOR THE START OF THE GAME *
* THIS MODULE IS EXECUTED ONLY WHEN THE REDO KEY IS PRESSED *
* LEAVES INTACT THE GAME AS IT WAS WHEN THE PLAYER GOT KILLED *
*****

```

```

RINITS LIM1 0
        MOV R11,@GPLRA        SAVE GPL RETURN ADDRESS FOR END
        LWPI MYWS             USE MY WORKSPACE AT >B300
        BL @SAVEPL           SAVE GPC AND CLEAR SPEECH
        LIM1 2
        MOVB @H00,@OBSPCH     ALLOW SPEECH PHRASE TO EXECUTE
        BL @CLRSCR           ZERO OUT THE SCORE
        LI R1,>2A             ASTERISK
        BL @WRTVCL           SIGNIFY NO START OF NEW GAME
        DATA SIT+>23B,1,R1LB
        BL @CLGAGE           SET THE WATER AND TEMP GAGES
        MOVB @H05,@CREWS      START WITH FIVE CREWS
        BL @SHWCRW           SHOW 5 MOONMINES AND SET CPU FLAG
        MOVB @HOC,@MOTION     ALLOW SPRITES TO MOVE
        B @MAINLP           GO TO MAIN PLAYING LOOP
        PAGE

```

```

*****
* THIS CODE DOES THE INITIS FOR THE START OF THE GAME *
* THIS MODULE IS EXECUTED ONLY WHEN THE BACK KEY IS PRESSED *
* STARTS THE GAME OVER JUST AS IF IF CAME FROM GROM. *
*****

```

```

BINITS LIM1 0
        MOV R11,@GPLRA        SAVE GPL RETURN ADDRESS FOR END
        LWPI MYWS             USE MY WORKSPACE AT >B300
        BL @SAVEPL           SAVE GPC AND CLEAR SPEECH
        LIM1 2
        BL @CLRSCR           ZERO OUT THE SCORE
        BL @WRTVCL
        DATA SIT+>23B,1,H20   SPACE OUT ANY ASTERISK
        MOVB @H00,@FLAG5      HAVEN'T HIT 5,000 POINTS YET
        MOVB @H00,@OBSPCH     ALLOW SPEECH PHRASE TO EXECUTE
        MOVB @H05,@CREWS      START WITH FIVE CREWS
        BL @SHWCRW           SHOW 5 MOONMINES AND SET CPU FLAG
        MOVB @HFF,@GAMELV     START WITH LEVEL -1 BECAUSE IT
        BL @NEWLVL           WILL BE INCREMENTED RIGHT AWAY
        BL @CLGAGE           SET THE WATER AND TEMP GAGES
        MOVB @SCROLD,@SCROLC  SET THE DELAY COUNTER FOR SCROLL
        MOVB @HOC,@MOTION     ALLOW SPRITES TO MOVE
        B @MAINLP           GO TO MAIN PLAYING LOOP
        PAGE

```

 * MAIN PLAYING LOOP FOR MOONMINE *

```

MAINLP CLR @SCNTIM          CLEAR SCREEN TIME OUT
      BL @INPUT            GO CHECK AND REACT TO INPUT
      BL @CDINC           TEST FOR BULLET HIT
      BL @DRWERS         GO SCROLL THE WALLS
      BL @MAKMN1        SEE IF ITS TIME FOR A MONSTER
      BL @INPUT            GO CHECK AND REACT TO INPUT
      BL @CDINC           TEST FOR BULLET HIT
      BL @DRWERS         GO SCROLL THE SCREEN
      BL @CHGMN1        SEE IF IT'S TIME TO CHANGE ITS PATTERN
      BL @INPUT
      BL @CDINC
      BL @DRWERS         GO SCROLL THE SCREEN
      BL @CHKWAT        SEE IF WATER NEEDS MADE, OR MOVED
  
```

```

*
      MOV @MKILED, R5     SEE HOW MANY WE'VE KILLED
      CB R5, @HOC        TWELVE?
      JL MAINND          NO, SO KEEP LOOPING
WHERE1 MOV @MN1FLG, R0   IS MONSTER 1 STILL ON SCREEN?
      JEQ WHERE2        NO, SO CHECK 2
      JMP NOGOON        YES, SO CAN'T ADVANCE YET
WHERE2 MOV @MN2FLG, R0   IS MONSTER 2 STILL ON SCREEN?
      JEQ GOON          NO, SO I CAN GO ON!! YEAH!!
NOGOON MOV @H00, @LETMON DON'T ALLOW ANY MORE MONSTERS
      JMP MAINND        AND KEEP LOOPING
GOON   BL @NEWLVL       GO TO NEW LEVEL
  
```

```

*
MAINND B @MAINLP        MAIN LOOP END
      PAGE
  
```

 * THIS IS A MODULE THAT WILL INCREMENT THE DEPTH LINES *
 * RETURN IS IN R13. *

```

DRWERS MOV R11, R13     SAVE RETURN
      SB @H01, @SCROLL  DECREASE SCROLL COUNTER
      JNE DRERRT       NO DONE, SO DON'T MOVE WALLS
      MOV @SCROLD, @SCROLLC  DONE, SO RESET DELAY COUNTER
TEST3  BL @VDPCLD      GO DRAW THEM BASED ON LOAD #
      AB @H01, @LDNO   INCREMENT LOAD #
      CB @LDNO, @H03   AT HALF-WAY POINT?
      JNE TEST7       NOT YET
      JMP SNDTNK
TEST7  CB @LDNO, @H07
      JNE TEST11
      JMP SNDTNK
TEST11 CB @LDNO, @H0B
      JNE TEST15
      JMP SNDTNK
TEST15 CB @LDNO, @H0F
      JLE DRERRT      PAST THE 16th LOAD?
      MOV @H00, @LDNO YES, SO JUST RETURN
      MOV @H00, @LDNO YES, SO START OVER
SNDTNK MOV @STRSD, R0   IS THERE A SOUND EXECUTING?
      JNE DRERRT     YES, SO NO NEW SOUND HERE
      BL @SOUND
      DATA TNKSND    EXECUTE TANK SOUND
DRERRT B *R13         RETURN TO CALLER
      PAGE
  
```

 * THIS MODULE SETS UP THE DATA TO ZERO OUT THE SCORE LOCATIONS*
 * USES REGISTER 15 TO SAVE THE RETURN LINKAGE *

```

*****
CLRSCR MOV R11,R15          SAVE RETURN LINKAGE
      LI VDPADD,SIT+>255    ADDRESS OF 'SCORE:'
      BL @WRTCL
      DATA 6,ZSCORE        PUT ALL ZEROES THERE
      B *R15                AND RETURN
*****
*   THIS MODULE SETS UP THE DATA TO ZERO OUT THE HIGH SCORE   *
*   LOCATIONS. RETURN LINK IS IN R15.                            *
*****
CLRHI  MOV R11,R15          SAVE RETURN LINKAGE
      LI VDPADD,SIT+>2B5    ADDRESS OF 'HIGH : '
      BL @WRTCL
      DATA 6,ZSCORE        PUT ALL ZEROES THERE
      B *R15
*****
*   THIS MODULE WRITES TO VDP REGISTER ONE AND ALLOWS ACTIVE   *
*   DISPLAY ON THE SCREEN.                                       *
*****
SCRNON MOV R11,R15          SAVE RETURN
      BL @VWTR              WRITE TO THAT REGISTER
      DATA >E2B1
      MOVB @HE2,@SVVDP1     AND PUT VALUE TO >83D4
      B *R15                AND RETURN
      PAGE
*****
*   THIS MODULE CLEARS THE TEMPERATURE GAGE AND FILLS THE WATER *
*   GAGE, THEN SETS THEIR POINTERS. RETURN IS IN R12           *
*****
CLGAGE MOV R11,R12          SAVE RETURN
      LI VDPADD,SIT+>22A    SIT START OF TEMP GAGE
      MOV R0,@TGAGPT        SET POINTER
      BL @WRTCL
      DATA 10,TGEMPT
      AI VDPADD,>0020        SIT START OF WATER GAGE
      BL @WRTCL
      DATA 10,WGFULL
      AI VDPADD,9           POINT TO END OF WATER GAGE
      MOV VDPADD,@WCAOPT    MOVE TO WATER GAGE POINTER
      MOVB @HOB,@WGCHAR     SET WATER GAGE CHARACTER
      MOVB @WGTIMV,@WGTIME  AND THE DELAY TIMER
      MOVB @SCROLD,@TGCOOL  AND DELAY TIMER FOR TEMP GAGE
      LI R1,>5100           81 GALLONS OF WATER
      MOVB R1,@GALWAT       AND SET THE CPU VARIABLE TO THAT
*** I'M ASSUMING THAT IF THIS ROUTINE IS CALLED, EITHER THE SHIP HAS BEEN
*** DESTROYED OR THE WATER HAS BEEN REPLENTISHED. THEREFORE, I MAY RESET
*** ALL FLAGS ASSOCIATED WITH THE WATER REPLENTISHING!!!
      CLR @WONSCN          ALSO GETS WBEENB (NOT ON SCREEN)
      MOVB @H00,@OBEENB    CAN ALLOW NEW OBJECT NOW
      MOVB @H00,@WTLOSS    ALLOW WATER TO DECREASE
      MOVB @H00,@OBJFLG    NO OBJECTS ON SCREEN ANYMORE
      B *R12                AND RETURN
      PAGE
*****
*   THIS MODULE DISPLAYS THE NUMBER OF CREWS LEFT. BECAUSE IS   *
*   CAN BE CALLED BY KILL ROUTINES, RETURN IS IN R14           *
*****
SHWCRW MOV R11,R14          SAVE RETURN
      MOVB @CREWS,R6        # OF CREWS REMAINING
      SRL R6,8              INTO LSB
      CI R6,5               MORE THAN 4 REMAINING?
      JLE SHOWT            NO, SO SHOW THEM

```

```

SHOWT  LI R6,>0005          YES, SO SHOW ONLY 4 FOR THE TIME BEING
        LI R0, 10          MULTIPLICAND
        MPY R0,R6          x10=OFFSET INTO TABLE
        LI R10,CWLFT1     POINT TO TOP CHARS
        A R7,R10          ADD OFFSET
        MOV R10,R9        PRESERVE IT
*
        LI VDPADD,SIT+>2A4 SIT ADDRESS FOR START OF CREWS
        LI RCOUNT, 10    8 BYTES TO MOVE
        MOV R10,RLOC      FROM THAT ADDRESS
        BL @GV            WRITE FROM GROM
*
        AI R9,60          POINTS TO BOTTOM CHARS
        AI VDPADD,>0020   POINTS TO NEXT ROW ON SCREEN
        LI RCOUNT, 10    8 BYTES TO MOVE
        MOV R9,RLOC      FROM THAT ADDRESS
        BL @GV            WRITE FROM GROM
*
NOSHOW B *R14            AND RETURN
        PAGE
*****
* THIS MODULE IS THE MINI_DRIVER TO GET INPUT FROM EITHER THE *
* KEYBOARD OR THE JOYSTICKS AND PERFORM THE NECESSARY ACTION. *
* RETURN LINKAGE IS REGISTER FOURTEEN                          *
*****
INPUT  MOV R11,R14        RETURN LINK
        MOV B @H00,@STOPFL SET INITIALLY TO NO INPUT
        CLR @KEYBRD       SCAN KEYBOARD ZERO
        MOV B @OBJFLO,@OBJFLO IS THERE ANY OBJECT ON THE SCREEN?
        JEQ BRDINC        NO, SO CAN'T SEND CREW MEMBER
        BL @SCANKY        YES, SO SEE IF WE WANT TO RETRIEVE IT
        CB @KEY,@H20      SPACE BAR PRESSED?
        JNE BRDINC        NO, SO CONTINUE ON
        MOV B @DEMOFL,R4  ARE WE IN DEMO MODE?
        JNE GOODBY        YES, SO RETURN TO TITLE SCREEN
        B @MAINL2        SWITCH GAME LOGIC
*
BRDINC AB @H01,@KEYBRD   INCREMENT KEY UNIT
        CB @KEYBRD,@H03  PAST SPLIT KEYBOARD?
        JNE BINCCT        NO, SO CONTINUE SCANNING
        B @INPTRT        YES, SO END THIS ROUTINE
BINCCT BL @SCANKY        GO GET INPUT
        CLR R3           CLR R3
        MOV B @KEY,R3    MOVE TO R3
        CI R3,>FF00      IS A KEY PRESSED?
        JEQ CHDEM1       NO, SEE IF DEMO WANTS TO MOVE
        JMP CHDEM2       YES, SEE IF WE GO BACK TO GROM
*****=> NO KEY DETECTED <*****
CHDEM1 MOV B @DEMOFL,R4  ARE WE IN DEMO?
        JEQ CHKJOY       NO, SO CHECK JOY AS USUAL
*
COMPKY SB @H01,@CMVDLY  -1 FROM COMPUTER MOVE DELAY
        JLT NEWDIR      TIME FOR NEW DIRECTION!
NONEWD MOV B @CMVDIR,R3 GET OLD DIRECTION
        JMP WATKEY      AND CONTINUE MOVING THAT WAY
NEWDIR BL @RANDNO       GET A RANDOM MOVE
        ANDI R10,7      MASK IT OUT
        CI R10,2        RAND # FOR FIRE?
        JEQ COKYCT      YES, SO DON'T REPEAT THIS VALUE
        CI R10,5        RAND # FOR FIRE?
        JEQ COKYCT      YES, SO DON'T REPEAT THIS VALUE
        MOV B @MONTIM+2,@CMVDLY SAVE OUT A DELAY VALUE FOR MOTION

```

```

COKYCT MOV B @DEMOKEY(R10),R3      WE HAVE THE COMPUTER'S MOVE
      MOV B R3,@CMVDIR             SAVE THE DIRECTION OUT
      JMP WATKEY
*****=> KEY DETECTED <=====
CHDEM2 MOV B @DEMOFL,R4           ARE WE IN DEMO MODE?
      JEQ WATKEY                   NO, SO PLAYER IS PLAYING
GOODBY B @GOGROM                  YES, SO RETURN TO GROM
*****=> DETERMINE WHAT KEY HAS BEEN HIT <=====
WATKEY MOV B @H00,@JOYFLG        NO JOYSTICK HIT HERE
      CI R3,>0B00                  PAUSE KEY?
      JEQ PAUSE
      CI R3,>0500                  UP KEY
      JEQ UP
      CI R3,>0000                  DOWN KEY
      JEQ DOWN
      CI R3,>0200                  LEFT KEY
      JEQ LEFT
      CI R3,>0300                  RIGHT KEY
      JEQ RIGHT
      CI R3,>0D00                  . OR V FIRE
      JEQ FIRE
      CI R3,>1200                  Q OR Y FIRE
      JEQ CHKJ1
***** WRONG KEY PRESSED AT THIS POINT
      JMP BRDINC                   TRY NEXT KEYBOARD
CHKJOY MOV @JOYY,@JOYY           COMPARE TO ZEROES
      JEQ BRDINC                   JOYSTICK NOT MOVED, SO CHECK NEXT KEYBOARD
CHKJ1  MOV B @H01,@JOYFLG        GOT TH' JOYSTICK FER' SURE
      MOV B @JOYY,R0              COMPARE TO ZERO
      JGT UP                       ARITH. > 0 (JOYSTICK MUST BE UP)
      JLT DOWN                     ARITH. < 0 (MUST BE DOWN)
      MOV B @JOYX,R0              COMPARE TO ZERO
      JGT RIGHT
      JLT LEFT
CHKJB  BL @SCANKY
      MOV B @KEY,R3               SAVE KEY
      SRL R3,8                    INTO LSB
      CI R3,>0012                  FIRE BUTTON?
      JEQ FIRE                     YES, SO FIRE
      B @BRDINC                   NO, SO CHECK NEXT KEYBOARD
PAGE

```

```

*****
* JUMPS MAY ONLY BE WITHIN 100 BYTES OF THE CALLER, SO THESE NEXT*
* FEW LINES ALLOW THOSE ROUTINES TO BE PLACED ANYWHERE IN THE PGM*
*****

```

```

UP      MOV B @H00,@PAIRFL        SET PAIR FLAG TO SIDE PAIR MOVED LAST
      SB @H01,@WGTIME            DECREASE DELAY TIMER FOR WATER
      JNE UPC                     NOT ZERO, SO DON'T DECREASE
      BL @DECRWG                 ZERO, SO DECREASE WATER GAGE
UPC     MOV B @HFF,@STOPFL        HAVE INPUT, SO NO STOPPING NOW
      LI R7,>DF00                UPWARD VELOCITY
      B @UPDOWN                  GOTO UP/DOWN SUBROUTINE
DOWN    MOV B @H00,@PAIRFL        SET PAIR FLAG TO SIDE PAIR MOVED LAST
      SB @H01,@WGTIME            DECREASE DELAY TIMER FOR WATER
      JNE DOWNC                  NOT ZERO, SO DON'T DECREASE
      BL @DECRWG                 ZERO, SO DECREASE WATER GAGE
DOWNC   MOV B @HFF,@STOPFL        HAVE INPUT, SO NO STOPPING NOW
      LI R7,>2000                DOWNWARD VELOCITY
      B @UPDOWN                  GOTO UP/DOWN SUBROUTINE
LEFT    MOV B @H01,@PAIRFL        PAIR FLAG TO VERT PAIR MOVED LAST
      SB @H01,@WGTIME            DECREASE DELAY TIMER FOR WATER
      JNE LEFTC                  NOT ZERO, SO DON'T DECREASE

```

LEFTC	BL @DECRWG MOV B @HFF, @STOPFL LI R7, >DF00 B @LFTRIT	ZERO, SO DECREASE WATER GAGE HAVE INPUT, SO NO STOPPING NOW LEFTWARD VELOCITY GOTO LEFT/RIGHT SUBROUTINE
RIGHT	MOV B @H01, @PAIRFL SB @H01, @WGTIME JNE RIGHTC BL @DECRWG	SET PAIR FLAG TO VERT PAIR MOVES LAST DECREASE DELAY TIMER FOR WATER NOT ZERO, SO DON'T DECREASE ZERO, SO DECREASE WATER GAGE
RIGHTC	MOV B @HFF, @STOPFL LI R7, >2000 B @LFTRIT	HAVE INPUT, SO NO STOPPING NOW RIGHTWARD VELOCITY GOTO LEFT/RIGHT SUBROUTINE
FIRE	MOV B @STOPFL, R0 JEG @GFIRE MOV B @HFF, @STOPFL	ARE THE GUNS MOVING WHILE FIRING? NO, SO DON'T RESET THE STOP FLAG YES, SO KEEP MOVING WHILE FIRING
@GFIRE	SB @H02, @WGTIME JGT FIRNOW BL @DECRWG	MORE WATER DUE TO FIRING IF > 0, DON'T LOSE ANY YET LOSE SOME WATER
FIRNOW	B @FIRES	GOTO FIRE SUBROUTINE
PAUSE	BL @PAUSES B @BRDINC	PAUSE SUBROUTINE AND NEXT KEY
	PAGE	

 * THIS IS THE RETURN POINT FOR ALL MOVING SUBROUTINES *

INPTRT	MOV B @STOPFL, R0 JNE INPUTR LI VDPADD, SVT	ARE WE SUPPOSED TO STOP? NO, SO JUST RETURN LOAD VDPADD
STOPGL	BL @WRTCL DATA 2, H0000 AI VDPADD, 4 CI VDPADD, SVT+>10 JL STOPGL	VDPADD ALREADY LOADED TWO BYTES OF ZEROES POINT TO NEXT SPRITES' VELOCITY AT FIFTH SPRITE? NO, SO LOOP AGAIN
INPUTR	MOV B @SHOTFL, R0 JNE IRTN SB @H01, @TGCOOL JNE IRTN MOV B @SCROLD, @TGCOOL MOV @TGAGPT, VDPADD DEC VDPADD CI VDPADD, SIT+>229 JEG IRTN LI R6, >0014 BL @WRTCL DATA 1, R6LB MOV VDPADD, @TGAGPT	SHOT BEEN FIRED? YES, SO DON'T COOL OFF, BABY! IS IT TIME TO COOL DOWN, YET? NO, SO DON'T COOL OFF YES, SO RESET COOL TIMER SET UP VDPADD WAS POINTING TO NEXT HEAT LOC PAST LEFT BORDER? YES, SO IT'S AT IT'S COOLEST! COOL CHARACTER (COOL CAT?)
IRTN	B *R14	FROM R6LB AND I WILL! RETURN TO CALLER
	PAGE	

 * THIS SUBROUTINE WILL SET THE VELOCITY TABLE TO MOVE UP OR DOWN *

UPDOWN	BL @EVENLR LI VDPADD, SVT BL @STOPAL MOV R7, R1 BL @WRTVCL DATA SVT+8, 2, MYWS+2 BL @WRTVCL DATA SVT+12, 2, MYWS+2 MOV B @JOYFLG, R3 JEG UDRT B @CHKJB	SEE IF WE ARE ALLOWED TO MOVE UP YES, SO STOP OTHER PAIR UPWARD VELOCITY OF 20 AND WRITE NEW SPRITE POSITION ALSO THE PARALLEL SPRITE DID WE USE JOYSTICK? NO, SO SCAN NEXT KEYBOARD SEE IF JOYSTICK WANTS TO FIRE NOW
UDRT	B @BRDINC	CHECK NEXT KEYBOARD

PAGE

* THIS SUBROUTINE WILL SET THE VELOCITY TABLE TO MOVE LEFT OR RIGHT *

LFTRIT BL @EVENTB SEE IF WE CAN MOVE THAT WAY
 LI VDPADD,SVT+8 YES, SO STOP OTHER PAIR
 BL @STOPAL
 SWPB R7
 MOV R7,R1 DECREMENT POSITION
 BL @WRTVCL
 DATA SVT+0,2,MYWS+2 AND WRITE NEW SPRITE POSITION
 BL @WRTVCL
 DATA SVT+4,2,MYWS+2 ALSO THE PARALLEL SPRITE
 MOVB @JOYFLG,R3 DID WE USE JOYSTICK?
 JEQ LRRT NO, SO SCAN NEXT KEYBOARD
 B @CHKJB YES, SEE IF JOYSTICK WANTS TO FIRE
LRRT B @BRDINC CHECK NEXT KEYBOARD
 PAGE

* THIS NEXT ROUTINE INSURES THAT THE SPRITES *
* ARE ON THE SAME COLUMN AND CHECK FOR BOUNDS *

EVENTB MOV R11,R12 SAVE RETURN
 BL @RDVCL INSURES TOP AND BOT ARE THE SAME
 DATA SAL+1,1,MYWS+2 X OF TOP INTO GR1
 MOV R1,R6 SAVE IT IN R6
 BL @WRTVCL
 DATA SAL+5,1,MYWS+2 MAKE BOT THE SAME
 SRL R6,8 INTO LSB OR R6
LFTBND CI R6,>0044 LEFT BOUNDARY
 JLE CHKTDR PAST IT, BUT CHECK DIR
RGTBND CI R6,>00AC RIGHT BOUNDARY
 JHE CHKBDR PAST IT, BUT CHECK DIR
 JMP ETBRTN IT'S O.K.
CHKTDR MOV R7,R7 TEST DIR FOR ZERO
 JGT ETBRTN
 JMP STOPTB
CHKBDR MOV R7,R7
 JLT ETBRTN
 JMP STOPTB
ETBRTN B *R12

* STOP THE TOP AND BOTTOM GUNS SUBROUTINE SETUP *

STOPTB LI VDPADD,SVT SPRITE 0 VELOCITY TABLE
 BL @STOPAL GO ZERO OUT THOSE TWO VELOCITIES
 MOVB @H00,@STOPFL AND SET TO STOP OTHER PAIR TOO
 MOVB @JOYFLG,R3 DID WE USE JOYSTICK?
 JEQ STBGR NO, SO SCAN NEXT KEYBOARD
 B @CHKJB SEE IF JOYSTICK WANTS TO FIRE, NOW
STBGR B @BRDINC AND CHECK NEXT KEY!!
 PAGE

* THIS NEXT ROUTINE INSURES THAT THE SPRITES *
* ARE ON THE SAME ROW AND CHECK FOR BOUNDS *

EVENLR MOV R11,R12 SAVE RETURN
 BL @RDVCL INSURES SIDES ARE THE SAME
 DATA SAL+0,1,MYWS+2 Y OF LFT INTO R1
 MOV R1,R6 SAVE IT IN R6
 BL @WRTVCL
 DATA SAL+12,1,MYWS+2 MAKE RGT THE SAME

```

SRL R6,8                MOVE SIDE POS INTO LSB OR R6
TOPBND CI R6,>000A      TOP BOUNDARY
JLE CHKRDR              PAST IT, SO STOP
BOTBND CI R6,>0066      BOT BOUNDARY
JHE CHKLDR              PAST IT, SO STOPRECTION FIRST
JMP ELRRTN              O.K.
CHKRDR MOV R7,R7        TEST DIR FOR ZERO
JGT ELRRTN
JMP STOPLR
CHKLDR MOV R7,R7
JLT ELRRTN
JMP STOPLR
ELRRTN B *R12
*****
* STOP THE SIDE GUNS SUBROUTINE SETUP *
*****
STOPLR LI VDPADD,SVT+8  SPRITE 2 VEL TABLE
BL @STOPAL              GO ZERO OUT THOSE TWO VELOCITIES
MOVB @H00,@STOPFL      AND SET TO STOP OTHER PAIR TOO
MOVB @JOYFLG,R3        DID WE USE JOYSTICK?
JEQ STLRT              NO, SO SCAN NEXT KEYBOARD
B @CHKJB               SEE IF JOYSTICK WANTS TO FIRE, NOW
STLRT B @BRDINC        CHECK NEXT KEY!!
PAGE
*****
* SUB TO WRITE 0 VELs TO DESIGNATED TWO SPRITES *
*****
STOPAL MOV R11,R10
BL @WRTCL              VDPADD ALREADY LOADED IN R0
DATA 2,H0000           ZEROES TO Y AND X VELs
AI VDPADD,4            POINT TO NEXT SPRITE OF THE PAIR
BL @WRTCL              ZEROES TO ITS Y AND X VELs
DATA 2,H0000           RETURN TO CALLER
B *R10
PAGE
*****
* FIRE ROUTINE (PLACES BULLETS ON THE SCREEN) *
*****
FIRES MOVB @SHOTFL,R0  ARE BULLETS ON SCREEN?
JNE FIRERT             YES, SO CAN'T REFIRE YET
MOVB @H01,@SHOTFL     NO, SO SET THE SHOT FLAG
*
MOVB @GAMELV,R8       GET THE GAME LEVEL
SRL R8,8              INTO A LSB
CI R8,20              ARE WE PAST LEVEL 20?
JHE TWOTIM            YES, SO JUMP 2 CHARS AT A TIME
*
ONETIM LI R9,1        ONE CHAR
JMP HEATLP
TWOTIM LI R9,2        TWO CHARS
*
HEATLP MOV @TGAGPT,VDPADD SET UP VDPADD
MOV VDPADD,R5         SAVE TEMPORARILY
LI R6,>0060           HOT CAT!!
BL @WRTCL
DATA 1,R6LB
INC VDPADD            CHANGE POINTER
CI VDPADD,SIT+>234   PAST RIGHT EDGE?
JNE TOSVPT           NO, SO JUST SAVE TMP. GAGE PNT.
TNKGON BL @WRTMSG     LASERS OVERHEATED MESSAGE
DATA 4
BL @SPSUB

```

```

DATA 7
B @LSTCRW
TGSVPT MOV VDPADD,@TGAGPT
DEC R9
JNE HEATLP
MOVB @SCROLD,@TGCOOL
*
MOVB @PAIRFL,@PAIRFL
JEG LRPREP
*
UDPREP LI R1,>9000
BL @PREP
JMP PUTUDS
*
LRPREP LI R1,>9400
BL @PREP
*
PUTLRS BL @RDVCL
DATA SAL+8,2,MYWS+10
BL @RDVCL
DATA SAL+12,2,MYWS+12
SB @H08,R6LB
AI R5,8
LI R7,>0064
LI R8,>009C
JMP PUTBUL
*
PUTUDS BL @RDVCL
DATA SAL+0,2,MYWS+10
BL @RDVCL
DATA SAL+4,2,MYWS+12
AI R5,>0800
SB @H08,R6
LI R7,>6400
LI R8,>9C00
*
PUTBUL BL @WRTVCL
DATA SAL+16,2,MYWS+10
BL @WRTVCL
DATA SAL+20,2,MYWS+12
*
BULVEL BL @WRTVCL
DATA SVT+16,2,MYWS+14
BL @WRTVCL
DATA SVT+20,2,MYWS+16
*
BL @SOUND
DATA FIRSND
*
FIRERT B @BRDINC
PAGE
EXIT
*****
* MODULE TO CHANGE THE PATTERN OF THE BULLETS AS DESIGNATED IN*
* REGISTER ONE MSB. RETURN LINKAGE IS IN R8 *
*****
PREP MOV R11,R8
BL @WRTVCL
DATA SAL+18,1,MYWS+2
BL @WRTVCL
DATA SAL+22,1,MYWS+2
B *R8
PAGE

```

```

YES, SO I OVERHEATED!!!!
SAVE TEMP. GAGE POINTER
LOOP AGAIN FOR HIGHER LEVELS?
YES
NO. RESET TO COOL TIMER

WHICH PAIR MOVED LAST
SIDES, SO PREPARE FOR SIDE SHOTS

CHARACTER >90
CHANGE PATTERN OF BULLETS
AND GO PUT BULLETS ON THE SCREEN

CHARACTER >94
CHANGE PATTERN OF BULLETS

GET THE POSITION OF THE SIDE GUNS
GET LEFT Y,X INTO R5
GET RIGHT Y,X INTO R6
SAME HERE
OFFSET JUST A LITTLE BIT
LEFT VELOCITY
RIGHT VELOCITY
AND PUT THE BULLETS ON THE SCREEN

GET THE POSITION OF THE VERT GUNS
GET TOP Y,X INTO R5
GET BOT Y,X INTO R6
OFFSET A LITTLE BIT
SAME HERE
TOP VELOCITY
BOT VELOCITY

THIS WILL PUT THE BULLETS ON THE SCREEN
AND REWRITE TO BULLET 1
AND REWRITE TO BULLET 2

LETS PUT IN THE VELOCITIES, NOW
HERE'S THE FIRST ONE FROM R7
HERE'S THE SECOND ONE FROM R8

EXECUTE FIRE SOUND

*****
SAVE RETURN LINKAGE
CHANGE PATTERN OF BULLETS
*TO WHATEVER WAS LOADED
*IN R1 BEFORE THE CALL
*TO THIS ROUTINE
RETURN

```

 * THIS ROUTINE IS CALLED FROM 'DRWERS' TO INCREMENT THE DEPTH LINES *
 * ONE PIXEL. RETURN IS IN R10. *

```

VDPCLD MOV R11,R10      * SAVE RETURN ADDRESS
      CLR R4            * CLEAR LOOP REG
      CLR R9            * CLEAR OFFSET WORK REG.
      MOVB @LDNO,R5     * GET CHAR LOAD NO.
      SRL R5,8          * INTO LSB
      CB @R5LB,@H07     * CK LOAD NUMBER SIZE
      JLE VTLD07        * BYPASS IF 1-8
VTLD07 SB @H08,@R5LB    * ADJUST LOAD NUMBER
      LI R3,GCHRLD     * LOAD REF ADDRESS NO
      BL @GSBR2        * GO FIND STARTING ADDR
      SLA R5,1          * MULT NUMBER X 2
      A R5,R3          * ADD OFFSET FOR ADDRESS
      BL @GSBR1        * GO TO GROM READ

VTLD05 LIM1 2
      LIM1 0
      MOVB @GRMRD,R7   * GET MSB OF LOAD ADDR 1-8
      SWPB R7          * MOVE TO LOW NYBBLE
      MOVB @GRMRD,R7   * GET LSB OF LOAD ADDR 1-8
      NOP
      MOVB @GRMRD,R9   * GET MSB OF LOAD ADDR 9-16
      SWPB R9          * MOVE TO LOW NYBBLE
      MOVB @GRMRD,R9   * GET LSB OF LOAD ADDR 9-16
      CB @LDNO,@H07    * CK LOAD NUMBER SIZE
      JLE VTLD06      * KEEP 1-8 ADDR
      MOV R9,R7        * PUT 9-16 ADDRESS INTO WORK

VTLD06 LIM1 2          *NEED SOME SPEED.
      LIM1 0          *FOR INTERRUPTS
      MOVB R7,@VDPWA  * SEND LSB OF VDP WRITE
      SWPB R7        * SET UP MSB
      MOVB R7,@VDPWA  * SEND MSB OF VDP WRITE
VTLD04 MOVB @GRMRD,R6  * CK FOR LOOP CODE
      CB R6,@HAA     * CK FOR >AA CODE
      JEQ VTLD00     * REPEAT CHARS( LOOP)
      CB R6,@HAB     * CK FOR >AB INTER LOAD ADDR.
      JEQ VTLD05
      CB R6,@HAD     * CK FOR EXIT CODE
      JEQ VTLD02     * EXIT ON >AD CODE
      MOVB R6,@VDPWD * WRITE CHAR TO VDP
      JMP VTLD04     * EXIT - GET NEXT CHAR
VTLD00 MOVB @GRMRD,R6  * GET LOOP COUNTER OR TERM BYTE
      SRL R6,8       * MOVE COUNT TO LOW BYTE
      MOVB @GRMRD,R8  * GET CHAR TO REPEAT
VTLD01 MOVB R8,@VDPWD
      DEC R6         * DEC COUNT
      JNE VTLD01    * LOOP
      JMP VTLD04    * GET NEXT CHARACTER
VTLD02 LIM1 2
      B *R10        * RETURN
PAGE
  
```

 * UPON MOVEMENT, THIS ROUTINE WILL DECREASE THE WATER GAGE AT *
 * A SET RATE. RETURN LINKAGE IS IN R10. *

```

DECRWG MOV R11,R10      SAVE RETURN
      MOVB @WTLOSS,@WTLOSS  ARE WE ALLOWED TO DECREASE THE WATER?
      JNE DECRWR        NO, SO JUST RETURN
      MOVB @WGTIMV,@WGTIME  RESET TIMER
      MOVB @W0CHAR,R5     GET CURRENT CHARACTER
  
```


SRL R5,8	*INTO LSB
INC R5	POINT TO NEXT CHAR
CI R5,>0010	PAST LAST CHARACTER?
JLE WWGCHR	NO, SO WRITE NEW WATER GAGE CHAR
*	
LI R5,8	NEW CHAR WILL BE THE FULL CHAR
MOV @WGAGPT,R6	DECREASE SIT ADDRESS OF CURR. WATER GAGE
DEC R6	
MOV R6,@WGAGPT	AND SAVE WHERE WE ARE
CI R6,SIT+>249	OUT OF WATER?
JH WWGCHR	NO, SO WRITE
BL @SPSUB	OUT OF WATER
DATA 17	
BL @WRTMSG	OUT OF WATER MESSAGE
DATA 14	
B @LSTCRW	YES, SO ONE TANK GONE!
*	
WWGCHR MOV @WGAGPT,VDPADD	SET UP VDP ADDRESS
BL @WRTCL	AND WRITE NEW WATER GAGE CHAR
DATA 1,R5LB	FROM R5 LOW BYTE
*	
MOVB @R5LB,@WGCHAR	SAVE NEW CHARACTER
SB @H01,@GALWAT	THERE GOES 1 GALLON OF WATER!!
DECWGR B *R10	AND RETURN
PAGE	

* THIS MODULE WILL PUT/MOVE WATER ON THE SCREEN. SETS A LOT OF*	
* FLAGS(SEE CPURAM+>58). RETURN IS IN R12. *	

CHKWAT MOV R11,R12	SAVE RETURN
MOVB @OBJFLG,R1	IS OBJECT ALREADY ON SCREEN?
JEQ CWATC	NO, SO MUST CREATE IT FIRST
B @MOVWAT	YES, SO JUST MOVE IT!
CWATC MOVB @GALWAT,R1	GET CURRENT # OF GALLONS
SRL R1,8	INTO LSB
CI R1,16	16 OR LESS?
JH CHKOBJ	NO, SO SEE IF WE CAN SEND OBJECT
*	
MOVB @WBEENB,R1	WATER ALREADY OFFERED TO THIS SHIP?
JNE NYRTRN	YES, SO DON'T OFFER IT AGAIN
*	
MOVB @OBSPCH,R0	DID WE ALREADY SAY MESSAGE?
JNE MSG2	YES, SO DON'T SAY IT AGAIN
BL @SPSUB	
DATA 14	
MSG2 MOVB @H01,@OBSPCH	PHRASE EXECUTED
BL @WRTMSG	COOLANT IS RUNNING LOW MESSAGE
DATA 11	
CLR R5	SEND WATER FLAG
JMP SENDIT	
*	
CHKOBJ CI R1,45	< 55?
JLE CHK50	YES
JMP NOOBJ	NO, SO NO OBJECT YET
CHK50 CI R1,30	> 45?
JHE SNDOBJ	YES, SO OBJECT CAN BE SENT
*	
NOOBJ B @CKWATR	CAN'T SEND ANYTHING
*	
SNDOBJ MOVB @DBEENB,R0	HAS OBJECT BEEN HERE BEFORE?
JNE NYRTRN	YES, SO NO OBJECT
MOVB @OBSPCH,R0	DID WE ALREADY SAY MESSAGE?

JNE MSG1	YES, SO DON'T SAY IT AGAIN
BL @SPSUB	
DATA 11	
MOV B @H01, @OBSPCH	PHRASE EXECUTED
MSG1 BL @WRTMSG	
DATA 8	OBJECT DETECTED
LI R5, 1	SEND OBJECT FLAG
*	
SENDIT MOV B @MN1FLG, R1	IS MONSTER 1 STILL THERE?
JNE NOTYET	YES, SO WAIT UNTIL HE'S GONE
MOV B @MN2FLG, R1	IS MONSTER 2 STILL THERE?
JNE NOTYET	YES, SO WAIT UNTIL HE'S GONE
JMP PUTWAT	NO MONSTERS, SO PUT THE WATER OUT
*	
NOTYET MOV B @H00, @LETMON	DON'T MAKE ANY MORE MONSTERS
NYRTRN B @CKWATR	AND EXIT THIS MODULE
*	
PUTWAT MOV B @H00, @LETMON	IF THERE WAS A MONSTER ON SCREEN, AND YOU
* SHOOT IT, IF IT WAS THE 12TH MONSTER, NEWLVL ALLOWS MONSTERS TO APPEAR	
* AGAIN. SO I NEED THE ABOVE COMMAND TO PREVENT MONSTERS FROM APPEARING	
* WHILE THE WATER IS COMING OUT.	
*	
MOV R5, R5	WHAT DO WE SEND?
JEQ PUTH20	WATER
*	
PUTOBJ BL @RANDNO	
SRL R10, 14	0, 1, 2
CI R10, 3	
JEQ PUTOBJ	CANNOT BE 3
MOV B @TREASC(R10), @R2LB	GOT THE COLOR
SLA R10, 8	x by 256
LI RLOC, TREAS	START OF TREASURES
A R10, RLOC	POINTS NOW TO A TREASURE
MOV B @H01, @OBEENB	THIS OBJECT HAS BEEN HERE NOW
JMP PUTIT	
*	
PUTH20 BL @SPSUB	
DATA 15	
BL @WRTMSG	
DATA 12	WATER AHEAD MSG
LI RLOC, WATER	GROM ADDRESS OF WATER CHARACTERS
MOV B @H01, @WONSCN	WATER ON SCREEN
MOV B @H01, @WTLOSS	DO NOT LOSE WATER
MOV B @H01, @WBEENB	THIS WATER HAS BEEN HERE NOW
*	
PUTIT CLR @WPIXLS	ALSO GETS WSIZCH WITH IT
MOV B @H00, @OBSPCH	CAN SAY PHRASE AGAIN
MOV RLOC, @WATPNT	SAVE OUT START OF CURRENT OBJECT
BL @OVCV	
DATA 32, SDL+>100	GET PATTERN INTO SDL
LI R1, >4400	HERE'S THE STARTING Y
*	
WATWHR BL @RANDNO	GET A RANDOM #
ANDI R10, >7F00	0-127
SRL R10, 8	INTO LSB
CI R10, 96	PAST 96?
JH WATWHR	YES, SO DO IT AGAIN
AI R10, 72	72-168 IS THE X VALUE
MOV B @RALB, @R1LB	WE NOW HAVE A RANDOM X
MOV B @WONSCN, R10	IS THIS WATER
JEQ MAKOBC	NO, SO SET TO PROPER COLOR
*	

MAK0BB LI R2, >A004
JMP WRTOBJ

CHAR AND BLUE COLOR

*
MAK0BC ORI R2, >A000

HAVE THE COLOR IN LOW BYTE,

*
WRTOBJ BL @WRTVCL
DATA SAL+>28, 4, MYWS+2
MOVB @H01, @OBJFLG

NOW THE WATER IS ON THE SCREEN
OBJECT IS NOW ON SCREEN

*
*
MOVWAT BL @RDVCL
DATA SAL+>28, 1, MYWS+2
SRL R1, 8
CI R1, 112
JHE DELWAT
INC R1
BL @WRTCL
DATA 1, R1LB
AB @H01, @WPIXLS
CB @WPIXLS, @H06
JL CKWATR
AB @H01, @WSIZCH
CB @WSIZCH, @H07
JH RESPIX
MOVB @WSIZCH, R5
SRL R5, 8
SLA R5, 5
MOV @WATPNT, RLOC

GET OBJECT Y
INTO R1 LSB
AT BOTTOM EDGEW?
YES, SO GET RID OF IT
NO, SO LET'S MOVE OBJECT

FINE WITH ME
MOVED ONE MORE PIXEL
MOVED OUR LIMIT?
NO, SO JUST RETURN
INC # OF PATTERN CHANGES
PAST ITS LIMIT?
YES, SO DO NOT CHANGE PATTERN
NEEDED FOR GROM ACCESS
INTO LSB
x BY 32 FOR EACH PATTERN
POINTS TO WHAT OBJECT IT CURRENTLY IS

*
MOV0BJ A R5, R4
BL @GVCV
DATA 32, SDL+>100
RESPIX MOVB @H00, @WPIXLS
B @CKWATR

NOW ADD THE OFFSET

AND GET THE NEW PATTERN
RESET THE PIXEL COUNTER
AND RETURN

*
DELWAT BL @WRTVCL
DATA SAL+>28, 1, HCO
MOVB @H00, @WONSCN
MOVB @H00, @WTLOSS
MOVB @H00, @OBJFLG
MOVB @H01, @LETMON
BL @GRMVPD
DATA BLANKM, 26, SIT+>1E3

NO LONGER ON SCREEN, BUT STILL WAS HERE
CAN LOSE WATER NOW!!
NO LONGER AN OBJECT ON SCREEN
CAN ALLOW MONSTERS NOW!!
BLANK OUT MESSAGES

*
CKWATR B *R12
PAGE

RETURN TO CALLER

* THIS MODULE IS THE PAUSE FEATURE INCLUDED IN ALL T. I. GAMES. *
* IT WILL STAY HERE UNTIL ANY KEY IS HIT AGAIN *

PAUSES MOV R11, R9
MOVB @MOTION, R2
MOVB @H00, @MOTION
MOVB @KEYBRD, R4

SAVE RETURN LINK
SAVE MOTION BYTE
AND SET TO NO MOTION
SAVE WHICH KEYBRD IT WAS IN

BL @SOUND
DATA SNDOFF
PAUSL1 BL @SCANKY
MOVB @STATUS, R0
JNE PAUSL1
CLR @KEYBRD

TURN OFF SOUNDS
GO CHECK FOR SOME INPUT
KEY STILL PRESSED (DEBOUNCE?)
YES, SO WAIT UNTIL HE LETS UP

PAUSL2 CLR @SCNTIM
BL @SCANKY

SCAN ON ZERO NOW
CLEAR SCREEN TIME-OUT
NOW GO CHECK FOR A KEY

```

MOV B @STATUS, R0          KEY PRESSED?
JEG PAUSL2                 NOT YET

*
PAUSL3 BL @SCANKY
CB @KEY, @HFF             NOW WAIT UNTIL HE RELEASES THAT ONE
JNE PAUSL3
MOV B R2, @MOTION        RESTORE MOTION
MOV B R4, @KEYBRD       RESTORE KEYBRD
B *R9                    LET'S GO ON WITH THE GAME
PAGE

*****
* THIS IS THE MODULE THAT DETERMINES IF IT IS TIME TO PUT A *
* MONSTER ON THE SCREEN. IF SO, IT WILL . RETURN IS IN R15. *
*****
MAKMN1 MOV R11, @TEMPRT   EMERG. SAVE RETURN LINK
*
MOV B @CRWOUT, R3        IS THE CREW MEMBER IN THE TUNNEL?
JNE CREW01              YES, SO DON'T USE MAIN LOOP1 LOGIC
*
BL @INPUT               CHECK FOR INPUT
BL @COINC              TEST FOR COINCIDENCE
BL @DRWERS             AND SCROLL
JMP CREW11
*
* IF A COINCIDENCE IS DETECTED, THE SCORE ROUTINE USES R15 AS A RETURN
* LINK. THEREFORE, I CANNOT PLACE @TEMPRT IN R15 AT THIS POINT!!!
*
CREW01 BL @INPUT2        CHECK FOR INPUT
BL @COINC              AND COINC TO MONSTER
BL @CSHOTB            CHECK SHOT BOUNDARY
BL @COINCC           AND COINC ON CREW MEMBER
*
CREW11 MOV B @LETMON, R0  AM I ALLOWED TO MAKE MONSTERS?
JNE MKM1CT            YES. GOTO MaKe Monster 1 Continue
B @MKMNRT            NO, SO LEAVE THIS SUBROUTINE
*
MKM1CT MOV B @MN1FLG, R0 IS MONSTER ON SCREEN?
JNE MAKMN2           NO, SO CHECK MONSTER #2
*
GTMNPT BL @GETMPT       GET A PATTERN
* UPON RETURN, RLOC CONTAINS PATTERN ADDRESS OFFSET
MOV RLOC, @MN1PNT    SAVE THE POINTER BEFORE GVV
BL @GVCV
DATA 32, SDL+>CO    WRITE PATTERN TO SDL FOR MON 1
GTMNCL BL @GETMCO       GO GET A COLOR
CB R2, @H01         IS THIS MONSTER DANGEROUS?
JEG MAK1BL         YES, SO SET FLAG TO SAY SO
MOV B @H00, @M1DANG NO, SO SET FLAG TO SAY SO
JMP PUTMON
MAK1BL MOV B @H01, @M1DANG THIS MONSTER CAN KILL YA'
*
PUTMON LI R1, >3870    STARTING Y, X
BL @WRTVCL         AND PUT MON ON SCREEN
DATA SAL+>18, 2, MYWS+2
BL @WRTVCL        ALONG WITH THE COLOR
DATA SAL+>18, 1, MYWS+4
*
BL @GETMVL         GET MONSTER VELOCITIES IN R1
*
BL @WRTVCL
DATA SVT+>18, 2, MYWS+2 NOW WRITE OUT THE VELOCITIES
*

```

MOV B @H00, @M1CTIM	CLEAR MONSTER ONE TIMING BYTE
MOV B @H10, @M1MTIM	SET MOVEMENT DELAY TIMER
MOV B @H01, @MN1FLG	WE'RE AT THE FIRST PATTERN
MOV B @H00, @M1HITS	NO HITS TO START WITH
*	
PAGE	
MAKMN2 MOV B @GAMELV, R5	GET THE GAME LEVEL
SRL R5, 8	INTO LSB
CI R5, 20	BEYOND LEVEL 20?
JHE MKM2CT	YES, SO WE CAN HAVE TWO MONSTERS
B @MKMNRT	NO, SO RETURN
*	
MKM2CT MOV B @CRWOUT, R3	IS THE CREW MEMBER IN THE TUNNEL?
JNE CREW22	YES, SO DON'T USE MAIN LOOP1 LOGIC
*	
BL @INPUT	
BL @CDINC	
BL @DRWERS	
JMP CREW02	
*	
CREW22 BL @INPUT2	INPUT
BL @CDINC	COINC ON MONSTER
BL @CSHOTB	CHECK BULLET BOUNDARY
BL @COINCC	AND ON CREW MEMBER
*	
CREW02 MOV B @MN2FLG, R0	IS MONSTER ON SCREEN?
JNE MKMNRT	NO, SO RETURN
*	
GTMNP2 BL @GETMPT	GET PATTERN
* UPON RETURN, RLOC CONTAINS OFFSET FOR MONSTER PATTERN	
MOV RLOC, @MN2PNT	SAVE THE POINTER BEFORE GVV
LI RCOUNT, 32	32 BYTES TO A
BL @GVCV	
DATA 32, SDL+>E0	WRITE PATTERN TO SDL FOR MON 2
GTMNC2 BL @GETMCO	GO GET HIS COLOR
CB R2, @H01	IS THIS MONSTER DANGEROUS?
JEQ MAK2BL	YES, SO SET FLAG TO SAY SO
*	
MOV B @H00, @M2DANG	NO, SO SET FLAG
JMP PUTMN2	
MAK2BL MOV B @H01, @M2DANG	THIS MONSTER DEFINITELY DANGEROUS
*	
PUTMN2 LI R1, >3870	STARTING Y, X
BL @WRTVCL	AND PUT MON ON SCREEN
DATA SAL+>1C, 2, MYWS+2	
BL @WRTVCL	ALONG WITH THE COLOR
DATA SAL+>1F, 1, MYWS+4	
*	
BL @GETMVL	GET MONSTER VELOCITIES IN R1
*	
BL @WRTVCL	NEW SPRITE VELOCITY
DATA SVT+>1C, 2, MYWS+2	AND WRITE FROM REGISTER ONE
*	
MOV B @H00, @M2CTIM	CLEAR MONSTER 2 TIMER BYTE
MOV B @H10, @M2MTIM	SET MOVEMENT DELAY TIMER
MOV B @H01, @MN2FLG	WE'RE AT THE FIRST PATTERN
MOV B @H00, @M2HITS	NO HITS TO START WITH
*	
MKMNRT MOV @TEMPRT, R15	NOW I MAY RESTORE THE RETURN LINK!!
B *R15	
PAGE	

```

* ROUTINE TO GET MONSTER PATTERN ADDRESS INTO RLOC FOR GRMVP. RETURN
* IS IS R13
*****
GETMPT MOV R11,R13          SAVE RETURN
GMPL   BL @RANDNO          GET A RANDOM NUMBER
      MOV R10,R6           AND PLACE IT IN R6
      SRL R6,12           NOW HAVE RANDOM 4 LSB
      CI R6,11            PAST # OF MONSTERS?
      JH GMPL             YES, SO GO AGAIN!
      LI R2,256           MULTIPLICATION FACTOR FOR GROM
      MPY R2,R6           POINTS TO A RANDOM MONSTER PATTERN
      LI RLOC,MONS       STARTING MONSTER GROM ADDRESS
      A R7,RLOC          ADD OFFSET TO GET THE MONSTER
      B *R13             RETURN TO CALLER

```

```

*****
* ROUTINE TO GET MONSTER COLOR INTO REGISTER TWO MSBYTE. RETURN
* IS IS R13
*****
GETMCO MOV R11,R13          SAVE RETURN
GMCOL  BL @RANDNO          GET A RANDOM NUMBER
      ANDI R10,>000F      0-15
      CI R10,>0009       LIGHT RED?
      JEQ GMCOL          CAN'T BE
      CI R10,>0006       DARK RED?
      JEQ GMCOL          CAN'T BE
      CI R10,>0008       MED RED?
      JEQ GMCOL          CAN'T BE
      CI R10,>0001       BLACK?
      JEQ MAKBLK         YES, SO SET DANGEROUS FLAG
      MOV R10,R10        TRANSPARENT
      JEQ MAKBLK
      CI R10,>000A       YELLOW
      JEQ MAKBLK
      CI R10,>000B       YELLOW
      JEQ MAKBLK
      MOV B @RALB,R2     GOT THE COLOR IN R2 MSB
      JMP GMCORT         AND RETURN
MAKBLK LI R2,>0100       FORCE BLACK
GMCORT B *R13           RETURN TO CALLER
      PAGE

```

```

*****
* ROUTINE TO GET MONSTER VELOCITIES INTO REGISTER ONE. RETURN
* IS IS R13
*****
GETMVL MOV R11,R13          SAVE RETURN
      MOV B @GAMELV,R6   GET THE CURRENT GAME LEVEL
      SRL R6,8           INTO LSB
MN1140 CI R6,40          AT OR BEYOND LEVEL 40?
      JL MN1120         NO SO CHECK FOR 20
      LI R2,40
      S R2,R6           DECREASE BY 40
      JMP MN1100
MN1120 CI R6,20          AT OR BEYOND LEVEL 20?
      JL MN1100         NO, SO WE'RE IN THE FIRST SET
      LI R2,20
      S R2,R6           DECREASE BY 20
MN1100 BL @RANDNO          GO GET A RANDOM NUMBER
      MOV R10,R8         AND SAVE IT OUT
      SRL R10,14        GET A #, 0, 1, 2, OR 3
      LI R5,MONVEL      START OF THE VELOCITIES
      SLA R6,2          x BY 4 (4 BYTES/VEL LEVEL)
      A R6,R5           POINT TO START OF VEL LEVEL

```

```

A R10,R5          RANDOM VEL WITHIN A VEL LEVEL
MOVW *R5,R1      GOT THE Y
S R10,R5         POINT TO START OF VEL LEVEL AGAIN
*
ANDI R8,>00C0
SRL R8,6         NEW RAND #
A R8,R5          RANDOM VEL WITHIN A VEL LEVEL
MOVW *R5,@R1LB  GOT THE X
*
GMLVRT B *R13    RETURN TO CALLER
PAGE
*****
* THIS IS THE MODULE THAT WILL CHANGE THE MONSTER'S PATTERN *
* IN ACCORDANCE WITH THE TIMER BYTE.      RETURN IS IN R15. *
*****
CHGMN1 MOV R11,@TEMPRT      EMERG RETURN LINK SAVE AREA
*
MOVW @CRWOUT,R3          IS THE CREW MEMBER IN THE TUNNEL?
JNE CREWC1              YES, SO DON'T USE MAIN LOOP1 LOGIC
*
BL @INPUT
BL @COINC              MAIN LOOP 1 LOGIC
BL @DRWERS
JMP CREWC3
* IF A COINC IS DETECTED, THE SCORE ROUTINE USES R15 AS A RETURN LINK.
* THEREFORE, I CANNOT RESTORE MY RETURN IN R15, YET!!
CREWC1 BL @INPUT2
BL @COINC              MAIN LOOP 2 LOGIC
BL @CSHOTB
BL @COINCC
*
CREWC3 MOVW @MN1FLG,R7   IS MONSTER ON SCREEN?
JEQ CHGMN2             NO, SO CHECK MONSTER 2
SRL R7,8              PUT MN1FLG INTO LSB OR R7
MOVW @MONTIM(R7),R8   LETS FIND OUT HOW LONG TO STAY AT THIS SIZE
CB @M1CTIM,R8        IS TIMER BYTE PAST OUR TIME LIMIT?
JL CHGVL1             NO, BUT CHANGE VELOCITY ANYWAY
INC R7               INCREMENT NUMBER OF CHANGES
CI R7,9              PAST IT'S LIMIT?
JEQ DELMN1           YES, SO DELETE MONSTER OFF SCREEN
MOVW @R7LB,@MN1FLG   NO, SO SAVE THE NEW MONSTER FLAG
CHGSZ1 MOV @MN1PNT,RLOC LOAD GROM ADDRESS WE'RE POINTING TO
AI RLOC,32           AND POINT TO NEXT PATTERN
MOV RLOC,@MN1PNT    SAVE OUT THE NEW MONSTER POINTER
BL @GVCV            AND GET THAT PATTERN
DATA 32,SDL+>C0     INTO VDP RAM
MOVW @H00,@M1CTIM   RESTART THE TIMING SEQUENCE
CHGVL1 SB @H01,@M1MTIM TIME TO MOVE, YET?
JNE CHGMN2          NO, SO CHECK #2
MOVW @H10,@M1MTIM  YES. RESET MOVEMENT DELAY TIMER
*
BL @GETMVL          GO GET A NEW VELOCITY
*
BL @WRTVCL          NEW SPRITE VELOCITY
DATA SVT+24,2,MYWS+2 AND WRITE FROM REGISTER ONE
JMP CHGMN2         CHECK MONSTER 2 NOW
*
DELMN1 BL @WRTVCL
DATA SAL+>10,1,HCO
CLR R1             ZERO OUT VELOCITIES
BL @WRTVCL
DATA SVT+>18,2,MYWS+2

```

MOV B @H00, @MN1FLG	RESET MONSTER ON SCREEN FLAG
MOV B @M1DANG, R5	WAS THIS MONSTER DANGEROUS?
JEQ CHGMN2	NO, SO JUST LET HIM GO
BL @SPSUB	
DATA 8	MONSTER DAMAGED SHIP
BL @WRTMSG	YES, SO PUT UP MESSAGE
DATA 5	
B @LSTCRW	YES, SO YOU JUST LOST A MOONMINE
* PAGE	
CHGMN2 EQU \$	
* MOV B @CRWOUT, R3	IS THE CREW MEMBER IN THE TUNNEL?
JNE CREWC2	YES, SO DON'T USE MAIN LOOP1 LOGIC
* BL @INPUT	
BL @COINC	MAIN LOOP 1 LOGIC
BL @DRWERS	
JMP CREWC4	
* CREWC2 BL @INPUT2	
BL @COINC	MAIN LOOP 2 LOGIC
BL @CSHOTB	
BL @COINCC	
* CREWC4 MOV B @MN2FLG, R7	IS MONSTER ON SCREEN?
JEQ CHMNRT	NO, SO RETURN
SRL R7, 8	PUT MN1FLG INTO LSB OR R7
MOV B @MONTIM(R7), R8	LET'S FIND OUT HOW LONG TO STAY AT THIS SIZE
CB @M2CTIM, R8	IS TIMER BYTE PAST OUR TIME LIMIT?
JL CHGVL2	NO, BUT CHANGE VELOCITY ANYWAY
INC R7	INCREMENT NUMBER OF CHANGES
CI R7, 9	PAST IT'S LIMIT?
JEQ DELMN2	YES, SO DELETE MONSTER OFF SCREEN
MOV B @R7LB, @MN2FLG	NO, SO SAVE THE NEW MONSTER FLAG
CHGSZ2 MOV @MN2PNT, RLOC	LOAD FROM ADDRESS WE'RE POINTING TO
AI RLOC, 32	AND POINT TO NEXT PATTERN
MOV RLOC, @MN2PNT	SAVE OUT THE NEW MONSTER POINTER
BL @CVCV	AND GET THAT PATTERN
DATA 32, SDL+>E0	INTO VDP RAM
MOV B @H00, @M2CTIM	RESTART THE TIMING SEQUENCE
CHGVL2 SB @H01, @M2MTIM	TIME TO MOVE, YET?
JNE CHMNRT	NO, SO EXIT THIS MODULE
MOV B @H10, @M2MTIM	YES. RESET MOVEMENT DELAY TIMER
* BL @GETMVL	GO GET NEW VELOCITIES
* BL @WRTVCL	
DATA SVT+>1C, 2, MYWS+2	NEW SPRITE VELOCITY
JMP CHMNRT	AND WRITE FROM REGISTER ONE
	AND RETURNTER 2 NOW
* DELMN2 BL @WRTVCL	
DATA SAL+>1C, 1, HCO	
CLR R1	ZERO OUT VELOCITIES
BL @WRTVCL	
DATA SVT+>1C, 2, MYWS+2	
MOV B @H00, @MN2FLG	RESET MONSTER ON SCREEN FLAG
MOV B @M2DANG, R5	WAS THIS MONSTER DANGEROUS?
JEQ CHMNRT	NO, SO JUST LET HIM GO
BL @SPSUB	
DATA 8	MONSTER DAMAGED SHIP
BL @WRTMSG	YES, SO PUT UP MESSAGE


```

DATA 5
B @LSTCRW                YES, SO YOU JUST LOST A MOONMINE
*
CHMNRT MOV @TEMPRT,R15   NOW I MAY RESORE MY RETURN LINKAGE
B *R15                   RETURN TO CALLER
PAGE
*****
* THIS MODULE IS CALLED WHENEVER THE MOONMINE HAS BEEN DES- *
* TROYED. NO RETURN LINKAGE IS NEEDED SINCE IT JUMPS TO MAINLP*
*****
LSTCRW MOVB @H00,@MOTION STOP AUTO MOTION
LI R1,>D000              GET RID OF BULS AND MONS
BL @WRTVCL
DATA SAL+>10,1,MYWS+2
BL @SOUND
DATA SNDOFF            TURN SOUNDS OFF
*
CLR @STOPFL            GUNS CAN'T KEEP MOVING!
MOVB @H00,@SHOTFL     NO SHOT AT THIS POINT
MOVB @H00,@MN1FLG     NO MONSTER 1
MOVB @H00,@MN2FLG     NO MONSTER 2 EITHER
MOVB @H00,@CRWOUT     NO CREW MEMBER THERE EITHER
MOVB @H00,@CSHOTF     NEITHER IS HIS SHOT!!
*
BL @DELALL
DATA SAL+>10,SVT+>10,0 DELETE FROM GUNS ON
*
BL @SOUND
DATA EXPSND           MAKE THE EXPLOSION SOUND
*
LI R5,4               4 GUNS TO COLOR
LI VDPADD,SAL+3      POINT TO GUNS
COLGUN BL @WRTCL
DATA 1,H06           MAKE GUNS RED AGAIN
AI VDPADD,4          POINT TO NEXT GUN
DEC R5               LOOP DONE?
JNE COLGUN           NO, SO KEEP GOING
*
LI R8,18              LOOP COUNTER
FLASHL BL @WRTVCL     FLASH WALLS RED
DATA CT+>18,6,WALLFC
LI R5,>0D00           DELAY LOOP
FDLY1 DEC R5
JNE FDLY1
BL @WRTCL
DATA 6,WALLNC
LI R5,>0D00           DELAY LOOP
FDLY2 DEC R5
JNE FDLY2
DEC R8
JNE FLASHL
*
BL @MSGDLY           WASTE SOME TIME
BL @SPSUB
DATA 10
BL @WRTMSG
DATA 7
SB @H01,@CREWS      CREW MEMBER LOST
BL @SHWCRW          LOST ONE TANK
BL @MSGDLY          DISPLAY # REMAINING
MOV @H01,@LETMON    WASTE SOME TIME
MOV @CREWS,R0       ALLOW MONSTERS TO APPEAR AGAIN!
                    ANY CREWS LEFT?

```

```

JNE KEEPON YES, SO KEEP ON TRUCKIN'
B @GOGROM RETURN TO GPL
KEEPON BL @SENDZY INTERMISSION ZYGONAUGHT
BL @CLGAGE RESET THE GAGES
BL @WRTMSG
DATA 10 PLEASE CONTINUE MESSAGE
BL @SPSUB *
DATA 13 EXECUTE "PRESS FIRE TO BEGIN"
BL @MSGDLY RESTORE MOTION
MOVB @HOC, @MOTION
BL @GRMVPD
DATA BLANKM, 26, SIT+>1E3 BLANK OUT MESSAGE AREA

```

```

*
B @MAINLP
PAGE

```

```

*****
* ROUTINE TO WASTE SOME TIME FOR MESSAGES AND SO FORTH. *
* RETURN IS IN R11. *
*****

```

```

MSGDLY LI R12, 4 COUNTER
LTTKL1 LI R8, >FFFF
LTTKL2 DEC R8 WASTE SOME TIME
JNE LTTKL2
DEC R12
JNE LTTKL1
B *R11

```

```

*****
* ROUTINE TO DELETE ALL SPRITES AND THEIR VELOCITIES. R5 HAS *
* START OF SAL, R6 HAS START OF SVT, AND R7 HAS LOOP COUNTER *
* RETURN LINK IS IN R14. *
*****

```

```

DELALL MOV *R11+, R5 SAL START
MOV *R11+, R6 SVT START
MOV *R11+, R7 COUNTER
MOV R11, R14 SAVE RETURN

```

```

*
LOSTLP MOV R5, VDPADD SET UP
BL @WRTCL
DATA 1, HCO DELETE SPRITE
MOV R6, VDPADD SET UP
BL @WRTCL
DATA 2, H0000 NO MOVING, BOYS!

```

```

*
AI R5, 4
AI R6, 4 POINT TO NEXT SPRITE
*
DEC R7 DECREMENT LOOP COUNTER
JNE LOSTLP NOT DONE, SO KEEP GOING

```

```

*
B *R14 RETURN TO CALLER
PAGE

```

```

*****
* WHEN THE PLAYER LOSES A TANK, THIS ROUTINE DOES A LITTLE *
* TAUNTING OF THE PLAYER. RETURN LINKAGE IS IN R15. *
*****

```

```

SENDZY MOV R11, R15 SAVE RETURN LINK
MOVB @DEMOFL, R0 ARE WE IN DEMO MODE?
JEQ ZYINDM NO, SO CONTINUE
B @SZRTRN YES, DON'T SEND ZYGO
ZYINDM BL @RANDNO GET RANDOM NUMBER
ANDI R10, >0700 0-7
SRL R10, 8 INTO LSB

```

```

JEG SZGOON
CI R10,3
JEG SZGOON
B @SZRTRN
SZGOON BL @WRTMSG
DATA 9
BL @SPSUB
DATA 12
BL @MSGDLY
LI RLOC, ZYGO
LI R9, ZYGOEY
MOV RLOC, R8
BL @GVCV
DATA 32, SDL+>C0
MOV R9, RLOC
BL @GVCV
DATA 32, SDL+>E0
LI R1, >3848
LI R2, >9804
BL @WRTVCL
DATA SAL+>18, 4, MYWS+2
LI R2, >9C08
BL @WRTVCL
DATA SAL+>1C, 4, MYWS+2
*
LI R10, 7
BL @SQUND
ZYLOOP DATA ZYCSND
BL @WRTVCL
DATA SIT+>E9, 14, REDCNT
AI VDPADD, 32
BL @WRTCL
DATA 14, REDCNT
BL @ZDELAY
DATA >1000
*
MOV R8, RLOC
AI RLOC, 32
MOV RLOC, R8
BL @GVCV
DATA 32, SDL+>C0
MOV R9, RLOC
AI RLOC, 32
MOV RLOC, R9
BL @GVCV
DATA 32, SDL+>E0
*
BL @WRTVCL
DATA SIT+>E9, 14, BLKCNT
AI VDPADD, 32
BL @WRTCL
DATA 14, BLKCNT
BL @ZDELAY
DATA >1000
*
DEC R10
JNE ZYLOOP
*
MOV R9, R0
BL @RANDNO
SRL R10, 15
AI R10, 19

```

```

0, SO SEND ZYGO
3?
YES, SO SEND ZYGO
NOT THIS TIME, FELLA!

ZYGONAUGHT APPROACHING MSG

AND WASTE SOME TIME
START AT PATTERN 0
SAME FOR THE EYES
SAVE THAT POINTER

USE MONSTER 1 AREA

USE MONSTER 2 AREA FOR EYES

LEFT SIDE OF BLACK
CHARACTER AND COLOR

FROM R1 AND R2
EYE PATTERN AND RED COLOR

FROM R1 AND R2

COUNTER FOR 7 MORE PATTERNS

ZYGONAUGHT COMING SOUND

FLASH CENTER RED
NEXT ROW DOWN

WASTE SOME TIME
BEFORE CHANGING

SAVED POINTER
POINT TO NEXT PATTERN
AND RESAVE
GET THE NEW PATTERN

SAVED POINTER FOR EYES
NEXT PATTERN
AND RESAVE

GET NEW EYE PATTERN

RESTORE TO BLACK
NEXT ROW DOWN

DECREMENT COUNTER
LOOP ISN'T OVER, YET!

SAVE ZYGO EYES PATTERN

0, 1
19, 20

```

MOV R10,@SPADR	GET A RANDOM VOICE PHRASE!!
MOV R0,R9	RESTORE ZYGO EYES POINTER
*	
ZLOOP2 LI R10,4	NEW LOOP COUNTER
MOV R8,RLOC	CLOSED MOUTH RIGHT NOW
AI RLOC,32	POINT TO OPEN MOUTH
BL @GVCV	
DATA 32,SDL+>C0	AND CHANGE PATTERN
BL @ZDELAY	AND WASTE TIME
DATA >4000	
MOV R8,RLOC	NOW CLOSE MOUTH AGAIN
BL @GVCV	
DATA 32,SDL+>C0	
BL @ZDELAY	AND WASTE SOME TIME
DATA >2000	
DEC R10	
JNE ZLOOP2	NOT FINISHED WITH LOOP, YET!
*	
**** AT THIS POINT, WE'RE POINTING AT THE CLOSED MOUTH PATTERN (#8)	
**** WE NEED TO CHANGE TO SIDE PATTERN, MOVE, THEN CHANGE TO	
**** CLOSED MOUTH PATTERN AGAIN BEFORE SHRINKING!!!	
ZYSIDE MOV R8,RLOC	POINTING AT CLOSED MOUTH
AI RLOC,64	POINT TO SIDE VIEW
BL @GVCV	AND CHANGE THE PATTERN
DATA 32,SDL+>C0	
MOV R9,RLOC	POINTING AT CLOSED EYE PATTERN
AI RLOC,64	POINT TO SIDE VIEW
BL @GVCV	AND CHANGE THE PATTERN
DATA 32,SDL+>E0	
*	
ZLOOP4 INC R1	GOING TO MOVE ZYGO
BL @WRTVCL	
DATA SAL+>18,2,MYWS+2	
BL @WRTVCL	
DATA SAL+>1C,2,MYWS+2	AND THE EYES, TOO
BL @ZDELAY	WASTE SOME TIME
DATA >0200	
CI R1,>38AB	AT RIGHT EDGE?
JL ZLOOP4	NO, SO KEEP MOVING
*	
MOV R8,RLOC	YES, SO BACK TO CLOSED MOUTH
BL @GVCV	
DATA 32,SDL+>C0	
MOV R9,RLOC	YES, SO BACK TO CLOSED MOUTH
BL @GVCV	
DATA 32,SDL+>E0	AND THE SAME FOR THE EYES
*	
LI R10,7	SEVEN PATTERNS TO DECREASE
LI R12,32	SUBTRACTING VALUE
BL @SOUND	
DATA ZYGSND	ZYGONAUGHT GOING SOUND
*	
ZLOOP3 MOV R8,RLOC	POINTING TO CLOSED MOUTH PATTERN
S R12,RLOC	NEXT PATTERNS DOWN
MOV RLOC,R8	AND RESAVE POINTER
BL @GVCV	GET NEW PATTERN
DATA 32,SDL+>C0	
MOV R9,RLOC	SAME FOR THE EYES
S R12,RLOC	
MOV RLOC,R9	
BL @GVCV	
DATA 32,SDL+>E0	

```

BL @ZDELAY
DATA >0650
DEC R10
JNE ZLOOP3
DECREMENT COUNTER
NOT FINISHED VANISHING YET
*
BL @WRTVCL
DATA SAL+>18,1,HCO
BL @WRTVCL
DATA SAL+>1C,1,HCO
SAME FOR THE EYES
*
SZRTRN B *R15
AND RETURN TO CALLER
*****
* ROUTINE TO WASTE SOME TIME FOR THE SENDZY ROUTINE. VALUE TO *
* DELAY IS IN R6. RETURN IS IN R11 *
*****
ZDELAY MOV *R11+,R6
COUNTER
ZDLOOP DEC R6
JNE ZDLOOP
B *R11
PAGE
*****
* THIS IS THE MODULE THAT TESTS THE COINCIDENCE BETWEEN MY *
* BULLETS AND THE MONSTERS. RETURN LINKAGE IS IN R13. *
*****
COINC MOV R11,R13
SAVE RETURN LINK
CO1LP LI R10,SAL+16
POINT TO FIRST BULLET
MOV @CRWOUT,R3
IS CREW MEMBER IN TUNNEL?
JEQ CO1LP2
NO, SO THIS IS A LASER CHECK
MOV @CSHOTF,R3
DID HE FIRE?
JEQ CO1EXT
NO, SO EXIT THIS MODULE
JMP CO1CON
YES, SO DO THE CHECK
*
CO1LP2 MOV @SHOTFL,R3
ARE BULLETS ON THE SCREEN?
JEQ COINRT
NO, SO RETURN
*
CO1CON MOV @MN1FLG,R3
IS MONSTER 1 ON THE SCREEN?
JEQ CO2LP
NO, SO CHECK MONSTER 2
SRL R3,8
THIS WILL BE A POINTER FOR TOL
MOV @MONTLX(R3),R1
NOW WE HAVE X TOL. LIMIT IN R1
MOV @MONTLY(R3),R2
NOW WE HAVE Y TOL LIMIT IN R2
MOV R10,R0
SET UP THE VDPADD
BL @RDCL
AND GET BULLET YX
DATA 2,MYWS+10
*INTO R5
BL @RDVCL
AND GET MONSTER YX
DATA SAL+>18,2,MYWS+14
*INTO R7
BL @CHKCOI
CHECK FOR COINCIDENCE
MOV R6,R6
DID WE GET IT?
JEQ CO2LP
NO, SO CHECK NEXT MONSTER
***** COINCIDENCE DETECTED ON MONSTER ONE
LI R9,>18
OFFSET INTO SVT,SAL
BL @OOTCOI
DELETE EVERYTHING!
CO1EXT B @COINRT
AND RETURN
*
PAGE
CO2LP EQU $
MOV @CRWOUT,R3
IS CREW MEMBER IN TUNNEL?
JEQ CO2LP2
NO, SO THIS IS A LASER CHECK
MOV @CSHOTF,R3
DID HE FIRE?
JEQ COINRT
NO, SO EXIT THIS MODULE
JMP CO2CON
YES, SO DO THE CHECK
*
CO2LP2 MOV @SHOTFL,R3
ARE BULLETS ON THE SCREEN?

```

```

      JEQ COINRT                                NO, SO RETURN
*
CU2CON MOVB @MN2FLG,R3                        IS MONSTER 2 ON THE SCREEN?
      JEQ NOCDIN                                NO, SO CHECK NEXT BULLET
      SRL R3,8                                  THIS WILL BE A POINTER FOR TOL
      MOV @MONTLX(R3),R1                        X TOL LIMIT IN R1
      MOV @MONTLY(R3),R2                        Y TOL LIMIT IN R2LIMIT
      MOV R10,R0                                SET UP THE VDPADD
      BL @RDCL                                  AND GET BULLET YX
      DATA 2,MYWS+10                           *INTO R5
      BL @RDVCL                                  AND GET MONSTER YX
      DATA SAL+>1C,2,MYWS+14                   *INTO R7
      BL @CHKCOI                                CHECK FOR COINCIDENCE
      MOV R6,R6                                  DID WE GET IT?
      JEQ NOCDIN                                NO, SO CHECK NEXT BULLET?
***** COINCIDENCE DETECTED ON MONSTER TWO!!!!
      LI R9,>1C                                  OFFSET FOR SAL,SVT
      BL @GOTCOI                                DELETE EVERYTHING!
      B @COINRT                                  AND RETURN TO CALLER
*
NOCDIN MOVB @CRWOUT,R3                        IS HE OUT IN THE TUNNEL?
      JNE COINRT                                YES, SO THERE AREN'T 2 BULLETS
      AI R10,4                                  NO, SO CHECK NEXT BULLET
      CI R10,SAL+>1B                            ARE WE NOW INTO THE MONS?
      JEQ COINRT                                YES, SO START WITH BULLET 1
      JMP CO1LP2                                  NO, SO CHECK NEXT BULLET
*
COINRT B *R13                                  RETURN TO CALLER
      PAGE
*****
* ROUTINE TO DETERMINE COINCIDENCE AFTER Y'S AND X'S ARE SET IN REGISTERS *
* RETURN IS IN R11.                                                                    *
*****
CHKCOI MOV R5,R6                              MAKE A COPY OF BULLETS
      SRL R5,8                                  PUT Y INTO LSB OF R5
      ANDI R6,>00FF                             PUT X INTO LSB OR R6
      MOV R7,R8                                  MAKE A COPY OF MONSTERS
      SRL R7,8                                  PUT Y INTO LSB OF R7
      ANDI R8,>00FF                             PUT X INTO LSB OF R8
      S R5,R7                                    GEY Y DIFFERENCE
      ABS R7                                    AND MAKE POSITIVE
      C R7,R2                                    DIFF > TOLERANCE Y
      JH SETTNO                                  YES, SO NO HIT HERE!
      S R6,R8                                    GEY X DIFFERENCE
      ABS R8                                    AND MAKE POSITIVE
      C R8,R1                                    DIFF > TOLERANCE? X
      JH SETTNO                                  YES, SO NO HIT HERE!
SETTCO LI R6,1                                  COINCIDENCE FLAG
      JMP CCOINR                                  AND RETURN
SETTNO CLR R6                                  NO COINCIDENCE HERE
CCOINR B *R11
      PAGE
*****
* IF A COINCIDENCE IS DETECTED, THIS MODULE WILL STOP, DELETE BULLETS AND *
* STOP, DELETE THE MONSTER THAT WAS HIT. RETURN IS IN R12.                            *
*****
GOTCOI MOV R11,R12
      BL @WRTVCL                                STOP THE BULLETS
      DATA SVT+16,2,H0000
      C *R0+,*R0+                               POINT TO NEXT BULLET
      BL @WRTCL
      DATA 2,H0000

```

* DELETB	BL @WRTVCL DATA SAL+16, 1, HCO C *R0+, *R0+ BL @WRTCL DATA 1, HCO MOVB @H00, @SHOTFL MOVB @H00, @CSHOTF	REWRITE FROM R1 POINT TO NEXT BULLET REWRITE FROM R1 TURN OFF SHOT FLAG AND CREW SHOT FLAG
* ADHIT	CI R9, >18 JNE ADHIT2	FIRST MONSTER HIT? NO
ADHIT1	AB @H01, @M1HITS CB @M1HITS, @HITLIM JEG HALTMN JMP WOUND1	ADD 1 TO IT'S HITS MET ITS LIMIT? YES, SO IT'S DEAD NO, SO IT'S WOUNDED
ADHIT2	AB @H01, @M2HITS CB @M2HITS, @HITLIM JEG HALTMN	ADD 1 TO IT'S HITS MET ITS LIMIT? YES, SO IT'S DEAD
WOUND2	MOVB @M2HITS, R2 JMP WOUNDC	GET # OF HITS AND CHANGE COLOR
WOUND1	MOVB @M1HITS, R2	GET # OF HITS
WOUNDC	SRL R2, 8 DEC R2 MOVB @MNVNDC(R2), R1 LI VDPADD, SAL+3 A R9, VDPADD BL @WRTCL DATA 1, MYWS+2	PLACE IN LSB 0 IS THE OFFSET GET THE NEW COLOR POINT TO THE SPRITE TABLE COLOR ADD THE OFFSET
TSTKIL	CI R9, >18 JNE NOKIL2	FIRST MONSTER? NOPE
NOKIL1	MOVB @MN1FLG, R5 JMP HITMON	NEED THIS FOR SCORING ADD THE POINTS
NOKIL2	MOVB @MN2FLG, R5 JMP HITMON	NEED THIS FOR SCORING ADD THE POINTS
* HALTMN	CLR R1 LI R0, SVT A R9, R0 BL @WRTCL DATA 2, MYWS+2	HAVE TO STOP THAT MONSTER! ADDRESS OF SVT WHICH MONSTER TO STOP AND STOP IT
* DELMN	LI R0, SAL A R9, R0 BL @WRTCL DATA 2, HCO AB @H01, @MKILED	ADDRESS OF SAL WHICH MONSTER WAS HIT AND DELETE MONSTER ONE MORE MONSTER KILLED!!!
* HITMN1	CI R9, >18 JNE HITMN2 MOVB @MN1FLG, R5 MOVB @H00, @M1HITS CLR @MN1FLG JMP HITMON	MONSTER 1 HIT? NO YES, SO GET ITS LAST SIZE RESET TO NO HITS CLEAR MONSTER ONE FLAG AND TIMER
HITMN2	MOVB @MN2FLG, R5 MOVB @H00, @M2HITS CLR @MN2FLG	GET MONSTER 2'S LAST SIZE RESET TO NO HITS CLEAR MONSTER TWO FLAG AND TIMER
HITMON	SRL R5, 8 SLA R5, 1 MOV @MONPNT(R5), R10 BL @SCORE BL @SOUND DATA HITSND	INDEX INTO LSB OF R5 2 BYTES TO A VALUE WHAT'S IT WORTH TO YA'? ADD TO SCORE AND MAKE MONSTER HIT SOUND

```

*
BL @RANDND
ANDI R10,>000F          SAY A REWARD MESSAGE?
CI R10,7
JNE GCRTN

*
MOV @SPADR,@SPADR      IS THERE A PHRASE EXECUTING?
JNE GCRTN              YES, SO DON'T INTERRUPT IT!!
SAYWAT BL @RANDND      GET RANDOM PHRASE
ANDI R10,>0300
SRL R10,8
CI R10,3                3?
JEG SAYWAT             YES, SO TOO HIGH
C *R10+,*R10+         ADD 4=4,5 OR 6
MOV R10,@SPADR        SAY A PHRASE

*
GCRTN B *R12           AND RETURN TO CALLER
PAGE

*****
* THIS IS THE MODULE TO ADD TO THE SCORE ROUTINE. THE VALUE TO*
* BE WRITTEN IN PASSED DOWN IN R10. RETURN LINKAGE IS IN R15 *
*****
SCORE MOV R11,R15      SAVE RETURN LINKAGE
CLR DIGTHB             CLEAR DIGIT BYTES
CLR CARYHB            CLEAR CARRY BYTES
LI VDPADD,SIT+>25A    LAST DIGIT SCREEN LOCATION
RDIGIT BL @RDCL
DATA 1,DIGTLB         GO GET THAT LAST DIGIT
MOV R10,R9            SAVE IT OUT
ANDI R10,>000F        GET ONE OF THE DIGITS FROM THE PASSED POINTS
A R10,DIGTHB          ADD THAT POINT TO OUR DIGIT
A CARYHB,DIGTHB      AND ANY CARRY FROM PREVIOUS ADD
CLR CARYHB            AND ERASE THE CARRY NOW
CB @DIGTLB,@H39      ASCII VALUE FOR DIGIT '9'
JLE WDIGIT           NO THERE YET, SO GO ON TO NEXT DIGITS
MOVB @H01,@CARYLB    WE'VE GOT A CARRY, NOW!
SB @HOA,@DIGTLB      MAKE THAT DIGIT A ZERO AGAIN
WDIGIT BL @WRTCL
DATA 1,DIGTLB        AND WRITE OUT THAT NEW DIGIT
MOV R9,R10           RESTORE ORIGINAL PASSED POINTS
SRL R10,4            GET NEXT PLACE OF TEN INTO LSB
DEC VDPADD           GOING TO WRITE/READ NEXT LEFT SCREEN SPOT
CI VDPADD,SIT+>255  AT FIRST SCREEN SCORE POSITION?
JHE RDIGIT          NO, SO CONTINUE

*
* TEST FOR EXTRA CREW MEMBER AT 5,000 POINTS
*
BL @RDVCL
DATA SIT+>257,1,MYWS+2  GET 1000'S DIGIT INTO R1
SRL R1,8             PUT INTO LSB
CI R1,>35            IS IT A '5'?
JL NOCREW           NO, SO NO NEW CREW
MOVB @FLAG5,R2      WAS IT ALREADY A '5'?
JNE SCORRT         YES, SO CAN'T GET NEW CREW

*
NEWCRW MOVB @H01,@FLAG5  IS IT A '5'
AB @H01,@CREWS      ONE MORE CREW MEMBER
BL @SHWCRW          SHOW HOW MANY YOU HAVE
BL @WRTMSG          SHOW MESSAGE
DATA 3
BL @SPSUB
DATA 3              EXTRA CREW MEMBER

```



```

        JMP SCORRT                                AND EXIT
*
NUCREW MOVB @H00,@FLAGS                          NO EXTRA CREW, BUT ELIGIBLE FOR ONE
*
SCORRT B *R15                                    RETURN TO CALLER
        PAGE
*****
* THIS IS THE MODULE TO PUT OUT THE GAME LEVEL NUMBER BASED *
* ON THE CPU RAM VARIABLE. IS IS AUTOMATICALLY INCREMENTED *
* ON EVERY CALL. RETURN IS IN R15. *
*****
NEWLVL MOV R11,@TEMPRT                            SAVE RETURN LINKAGE
        AB @H01,@GAMELV                          ADD ONE TO CURRENT GAME GAME LEVEL
        JEQ NWLVGO                                ZERO, SO NO BONUS POINTS
        BL @WRTMSG                                ADVANCE TO NEW LEVEL MSG
        DATA 1
        BL @SPSUB
        DATA 1
        LI R10,>1000                              ONE THOUSAND POINTS
        BL @SCORE                                ADD TO SCORE
NWLVG0 MOV @TEMPRT,R15                          RESTORE RETURN
        LI R2,>3C00                                PAST 60th LEVEL?
        CB @GAMELV,R2
        JL NEWLCT                                NO, SO CONTINUE ON
        LI R2,>2800                                RESET TO players LEVEL 41
        MOVB R2,@GAMELV
NEWLCT CLR R4                                    PREPARE FOR DIVIDE
        MOVB @GAMELV,R5                          MOVE GAME LVL TO R5
        LI R8,>0030                                ASCII BIAS
        SRL R5,8                                  INTO LSB (0=PLAYERS LEVEL 1)
        INC R5                                    OFFSET IS ONE
        LI R1,10                                  DIVISOR
        DIV R1,R4                                 / BY 10, REMAIN. IN R5
        A R8,R4                                    ASCII OFFSET
        MOVB @R4LB,R7                            O.K. NOW I HAVE THE 10'S
        A R8,R5                                    ADD ASCII OFFSET
        MOVB @R5LB,@R7LB                         NOW I'VE GOT A TWO DIGIT #
        BL @WRTVCL
        DATA SIT+>24A,2,MYWS+14                FROM R7
        MOVB @H00,@MKILED                        NO MONSTERS KILLED IN NEW LEVEL
        MOVB @OBJFLG,@OBJFLG                    IS OBJECT ON THE SCREEN?
        JNE NEWSCC                               YES, SO STILL CAN'T ALLOW MONSTERS
        MOVB @H01,@LETMON                       AND ALLOW MONSTERS, NOW!
*
NEWSCC BL @CHGSCS                                GO CHANGE SCREEN COLOR
        BL @CHGHTL                                GO CHANGE HIT LIMIT
NWLVRT B *R15                                    RETURN TO CALLER
        PAGE
*****
* THIS IS THE MODULE THAT CHANGES THE SCREEN COLOR. VALUES CAN*
* BE PASSED IN R6. . RETURN LINKAGE IS R14 *
*****
CHGSCS MOV R11,R14                              RETURN LINK
        CLR R4                                    PREPARE FOR DIVIDE
        MOVB @GAMELV,R5                          GET CURRENT GAME LEVEL
        SRL R5,8                                  *INTO LSB
        LI R1,3                                    DIVISOR
        DIV R1,R4                                 REMAINDER IS IN R5
        LI WLOC,>0087                              WRITE FLAG TO VDP R7
        MOVB @LVLCLR(R5),WLOC                    VALUE TO BE WRITTEN IN MSB
SCLRGO BL @VWTRW                                SEND VALUE FIRST
        B *R14                                    AND RETURN

```

```

*****
* THIS IS THE MODULE THAT CHANGES THE CPURAM VARIABLE 'HITLIM'*
* DEPENDING ON SCREEN LEVEL. RETURN IN IN R14. *
* ALSO CHANGES VARIABLE 'SCROLD' FOR SCROLL DELAY VALUE. *
* ALSO CHANGES VARIABLE 'WGTIMV' FOR WATER GAGE DELAY VALUE. *
*****
CHGHTL MOV R11,R14          RETURN LINK
      CLR R4              PREPARE FOR DIVIDE
      MOVB @GAMELV,R5     GET CURRENT GAME LEVEL
      SRL R5,8           *INTO LSB
      LI R1,20          DIVISOR
      DIV R1,R4          DIVIDEND IN R4, REMAIND. IN R5
      MOVB @MHITLM(R4),@HITLIM SET THE NUMBER OF HITS REQUIRED
      MOVB @SCROLL(R5),@SCROLD AND THE SCROLL DELAY VALUE
      LI R4,19          TWENTY LEVELS TO A SECTION(0-19)
      S R5,R4           GET THE DIFFERENCE OF CURRENT
      SLA R4,2          x BY 4 (4 MONS VELS TO A TABLE)
      MOVB @MONVEL(R4),@WGTIMV SET THE TIMER VALUE
      AB @H10,@WGTIMV
* ON THE ABOVE STATEMENTS, I USED THE MONSTER VELOCITY TABLE IN REVERSE
* ORDER TO GET THE WATER GAGE DECREMENT VALUE.
      B *R14          AND RETURN
*****
* THIS IS THE MODULE THAT GENERATES A RANDOM NUMBER. IS IS *
* PASSED IN REGISTER 10. RETURN LINKAGE IS R11 *
*****
RANDNO LI R9,28645
      MPY @SEED,R9
      AI R10,31417
      MOV R10,@SEED      MAKE A NEW SEED
      B *R11          AND RETURN TO CALLER
      PAGE
*****
* WHEN THE GAME IS OVER, CONTROL PASSES HERE TO RETURN CONTROL*
* TO THE GPL ASSEMBLER. *
*****
GOGROM BL @GRMVDP
      DATA BLANKM,26,SIT+>1E3
      MOVB @DEMOFL,R3    ARE WE IN DEMO MODE?
      JNE INDEMO        YES, SO DELETE ALL SPRITES
OUTDMO BL @DELALL
      DATA SAL+>10,SVT+>10,8  EVERYTHING BUT THE GUNS
      JMP GOAWAY
INDEMO BL @DELALL      CANCEL ALL SPRITES
      DATA SAL,SVT,12
      BL @SOUND          TURN OFF ALL SOUNDS
      DATA SNDOFF
GOAWAY LIM1 0          DISABLE INTERRUPTS
      LWPI CPLWS        LOAD BACK GPL WORKSPACE
      MOVB @GPCSAV,@GRMWA RETURN ADDRESS
      MOVB @GPCSAV+1,@GRMWA RETURN ADDRESS
      MOV @GPLRA,R11    RESTORE RETURN ADDRESS
      B *R11          AND RETURN TO GPL
      PAGE
*****
* MAIN PLAYING LOOP FOR TUNNEL *
* SCENARIO *
*****
MAINL2 BL @SOUND      TURN OFF ALL SOUNDS, FIRST
      DATA SNDOFF
      BL @PTCREW      PUT THE CREW MEMBER IN THE TUNNEL
MAIN2 CLR @SCNTIM     DON'T WANT THE SCREEN GOING BYE BYE

```

BL @INPUT2	CHECK FOR SOME INPUT AND REACT TO IT
BL @COINC	CHECK FOR COINCIDENCE ON SHOT
BL @CSHOTB	CHECK FOR BOUNDARY ON BULLET
BL @COINCC	CHECK IF MONSTER HIT CREW MEMBER

*		
MOV B @OBJFLG,R3	IS THERE AN OBJECT ON THE SCREEN?	
JEQ M2CON1	NO, SO OBJECT HAS BEEN PICKED UP	
BL @RANDNO	GET A RANDOM #	
ANDI R10,>0F00	0-15	
SRL R10,8		
CI R10,7	CAN I SEND A MONSTER?	
JNE M2CON1	NO	
MOV B @H01,@LETMON	YES, SO ALLOW MONSTERS	
BL @MAKMN1	MAKE ONE APPEAR	
MOV B @H00,@LETMON	AND ALLOW NO MORE FOR NOW	

*		
M2CON1	BL @INPUT2	CHECK FOR SOME INPUT AND REACT TO IT
	BL @COINC	CHECK FOR COINCIDENCE ON SHOT
	BL @CSHOTB	CHECK FOR BOUNDARY ON BULLET
	BL @COINCC	SEE IF MONSTER HIT CREW MEMBER
	BL @CHGMN1	CHANGE MONSTER'S PATTERN
M2END	B @MAIN2	CYCLE THROUGH AGAIN
	PAGE	

 * THIS ROUTINE PUTS THE CREW MEMBER OUT IN THE TUNNEL. *
 * RETURN LINK IS R15. *

PTCREW	MOV R11,R15	SAVE RETURN LINK
	LI R5,4	COUNTER
	LI VDPADD,SAL+3	POINT TO COLOR
RIDGUN	BL @WRTCL	
	DATA 1,H00	MAKE GUN INVISIBLE
	AI VDPADD,4	POINT TO NEXT GUN
	DEC R5	LOOP DONE?
	JNE RIDGUN	NO, SO KEEP GOING

*		
	BL @GRMVDP	GET CREW UP PATTERN
	DATA CREWU,32,SDL+>140	USE CHARACTER >A8
	LI R1,>6760	ROW AND COL
	LI R5,5	LOOP COUNTER
PCLP1	LI R2,>A806	CHAR AND COLOR
	BL @WRTVCL	
	DATA SAL+>20,4,MYWS+2	PUT CREW MEMBER ON THE SCREEN
	BL @ZDELAY	
	DATA >1800	AND WASTE SOME TIME
	LI R2,>A800	TRANSPARENT NOW
	BL @WRTVCL	
	DATA SAL+>20,4,MYWS+2	
	BL @ZDELAY	
	DATA >1800	
	DEC R5	
	JNE PCLP1	NOT DONE FLASHING

*		
	LI R2,>A806	
	BL @WRTVCL	
	DATA SAL+>20,4,MYWS+2	NOW HE'S ON THE SCREEN
	MOV B @H01,@CRWOUT	AND SET THE FLAG TO SAY SO
	MOV B @H0D,@CLSLIM	12 SHOTS AND COUNTING

*		
PCRTRN	B *R15	AND RETURN TO CALLER
	PAGE	

```

* THIS ROUTINE IS THE INPUT DRIVER FOR THE TUNNEL SCENE.      *
* RETURN LINK IS R14.                                         *
*****
INPUT2 MOV R11,R14          SAVE RETURN LINK
      MOVB @H00,@STOPFL    SET INITIALLY TO NO INPUT
      CLR @KEYBRD         SCAN KEYBOARD ZERO
*
      MOVB @OBJFLG,R3      IS THE OBJECT STILL ON THE SCREEN?
      JNE BINC2           YES, SO CANNOT RETURN TO SHIP
*
      MOVB @MN1FLG,R3     IS MONSTER 1 STILL ON THE SCREEN?
      JNE BINC2           YES, SO CANNOT RETURN TO SHIP
*
      MOVB @MN2FLG,R3     IS MONSTER 2 STILL ON THE SCREEN?
      JNE BINC2           YES, SO CANNOT RETURN TO SHIP
*
      BL @SCANKY          GET SOME INPUT
      CB @KEY,@H20        SPACE BAR?
      JNE BINC2          NO
      B @RTCREW          YES, SO RETURN CREW MEMBER TO SHIP
*
BINC2 AB @H01,@KEYBRD      INCREMENT KEY UNIT
      CB @KEYBRD,@H03    PAST SPLIT KEYBOARD?
      JEQ INP2RT        YES, SO RETURN
      BL @SCANKY        GO GET A KEY
      CLR R3
      MOVB @KEY,R3
      CI R3,>FF00
      JEQ CK2JOY
*
WTKEY2 MOVB @H00,@JOYFLG  NO JOYSTICK HIT HERE
      CI R3,>0B00
      JEQ PAUSEC        PAUSE KEY
      CI R3,>0500
      JEQ UPCR          UP KEY
      CI R3,>0000
      JEQ DNCR          DOWN KEY
      CI R3,>0200
      JEQ LEFTCR       LEFT KEY
      CI R3,>0300
      JEQ RIGTCR       RIGHT KEY
      CI R3,>0D00
      JEQ FIRECR
      CI R3,>1200
      JEQ CHKJ2        Q OR Y
      JMP BINC2        WRONG KEY HIT
CK2JOY MOV @JOYY,@JOYY    COMPARE TO ZERO
      JEQ BINC2        JOYSTICK NOT MOVED, SO NEXT KEYBRD
CHKJ2  MOVB @H01,@JOYFLG  GOT THE JOYSTICK THIS TIME
      MOVB @JOYY,@JOYY    SEE FOR UP OR DOWN
      JGT UPCR
      JLT DNCR
      MOVB @JOYX,@JOYX    RIGHT OR LEFT
      JGT RIGTCR
      JLT LEFTCR
CHKJB2 BL @SCANKY
      MOVB @KEY,R3
      SRL R3,8
      CI R3,>0012
      JEQ FIRECR
      B @BINC2
PAGE

```

```

*****
* JUMPS MAY ONLY BE WITHIN 100 BYTES OF THE CALLER, SO THESE NEXT*
* FEW LINES ALLOW THOSE ROUTINES TO BE PLACED ANYWHERE IN THE PGM*
*****
LEFTCR B @LEFT2          LEFT ROUTINE
- RIOTCR B @RIGHT2       RIGHT ROUTINE
UPCR   B @UP2            UP ROUTINE
DNCR   B @DOWN2         DOWN ROUTINE
FIRECR B @FIRE2         FIRE ROUTINE
PAUSEC B @PAUSE2        PAUSE ROUTINE
*
*****
* THIS IS THE RETURN POINT FOR ALL MOVING SUBROUTINES      *
*****
INP2RT MOVB @STOPFL,R0   ARE WE SUPPOSED TO STOP?
      JNE I2RTN          NO, SO JUST RETURN
      LI VDPADD,SVT+>20  LOAD VDPADD
      BL @WRTCL          VDPADD ALREADY LOADED
      DATA 2,H0000      TWO BYTES OF ZEROES
      BL @GRMVDP
      DATA CREWU,32,SDL+>140 RESTORE TO STANDING POSITION
      MOVB @H00,@CRWDIR  JUST STANDING STILL, FOLKS
*
I2RTN  B *R14           RETURN TO CALLER
      PAGE
*****
* THIS IS THE ROUTINE TO MOVE THE CREW MEMBER LEFT. CRWDIR IS SET*
* TO 1.                                                    *
*****
LEFT2  MOVB @HFF,@STOPFL HAVE INPUT, SO NO STOPPING NOW
      MOVB @CRWDIR,R5    ARE WE GOING LEFT?
      CB R5,@H01
      JEQ GOLEFT        YES, SO ALL POINTERS ARE SET
      MOVB @H01,@CRWDIR GOING LEFT NOW
      MOVB @HFF,@CPCNTR START AT FIRST PATTERN
      MOVB @H01,@CPINC  AND GOING UP
      MOVB @H01,@CANDLY WILL BE DECREMENTED RIGHT AWAY!!
GOLEFT LI R6,CREWL      START ADDRESS OF LEFT CHARS
      MOV R6,@CCRMPPT   FOR GRMVDP
      BL @CCPATT        CHANGE THE PATTERN
      LI R7,>DFOO       LEFTWARD VELOCITY
      LI R6,CRLBND      LEFT BOUNDARY TABLE
      B @LR2            GOTO LEFT/RIGHT SUBROUTINE
      PAGE
*****
* THIS IS THE ROUTINE TO MOVE THE CREW MEMBER RIGHT. CRWDIR IS *
* SET TO 2.                                                    *
*****
RIGHT2 MOVB @HFF,@STOPFL HAVE INPUT, SO NO STOPPING NOW
      MOVB @CRWDIR,R5    ARE WE GOING RIGHT?
      CB R5,@H02
      JEQ GORIGT       YES, SO ALL POINTERS ARE SET
      MOVB @H02,@CRWDIR GOING LEFT NOW
      MOVB @HFF,@CPCNTR START AT FIRST PATTERN
      MOVB @H01,@CPINC  AND GOING UP
      MOVB @H01,@CANDLY WILL BE DECREMENTED RIGHT AWAY!!
GORIGT LI R6,CREWR      START ADDRESS OF RIGHT CHARS
      MOV R6,@CCRMPPT   FOR GRMVDP
      BL @CCPATT        CHANGE THE PATTERN
      LI R7,>2000       RIGHTWARD VELOCITY
      LI R6,CRRBND      CREW RIGHT BOUNDARY TABLE
      B @LR2            GOTO LEFT/RIGHT SUBROUTINE

```

PAGE

* THIS IS THE ROUTINE TO MOVE THE CREW MEMBER UP. CRWDIR IS SET *
* TO 3. *

UP2 MOV B @HFF,@STOPFL HAVE INPUT, SO NO STOPPING NOW
 MOV B @CRWDIR,R5 ARE WE GOING UP?
 CB R5,@H03
 JEG Goup YES, SO ALL POINTERS ARE SET
 MOV B @H03,@CRWDIR GOING LEFT NOW
 MOV B @HFF,@CPCNTR START AT FIRST PATTERN
 MOV B @H01,@CPINC AND GOING UP
 MOV B @H01,@CANDLY WILL BE DECREMENTED RIGHT AWAY!!
Goup LI R6,CREWU START ADDRESS OF UP CHARS
 MOV R6,@CGRMPT FOR GRMVDP
 BL @CCPATT CHANGE THE PATTERN
 LI R7,>F000 UPWARD VELOCITY
 LI R6,CRUBND CREW UP BOUNDARY TABLE
 B @UD2 GOTO UP/DOWN SUBROUTINE
PAGE

* THIS IS THE ROUTINE TO MOVE THE CREW MEMBER DOWN. CRWDIR IS *
* SET TO 4. *

DOWN2 MOV B @HFF,@STOPFL HAVE INPUT, SO NO STOPPING NOW
 MOV B @CRWDIR,R5 ARE WE GOING DOWN?
 CB R5,@H04
 JEG GODWN YES, SO ALL POINTERS ARE SET
 MOV B @H04,@CRWDIR GOING DOWN NOW
 MOV B @HFF,@CPCNTR START AT FIRST PATTERN
 MOV B @H01,@CPINC AND GOING UP
 MOV B @H01,@CANDLY WILL BE DECREMENTED RIGHT AWAY!!
GODWN LI R6,CREWD START ADDRESS OF RIGHT CHARS
 MOV R6,@CGRMPT FOR GRMVDP
 BL @CCPATT CHANGE THE PATTERN
 LI R7,>1000 DOWNWARD VELOCITY
 LI R6,CRDBND CREW DOWN BOUNDARY TABLE
 B @UD2 GOTO UP/DOWN SUBROUTINE
PAGE

* THIS IS THE FIRE ROUTINE. THIS CAUSES A BULLET TO EMERGE FROM *
* THE CREW MEMBER'S GUN. *

FIRE2 MOV B @STOPFL,@STOPFL IS THE CREW MEM MOVING?
 JEG GFIRE2 NO, SO DON'T RESET THE STOP FLAG
 MOV B @HFF,@STOPFL YES, SO KEEP MOVING WHILE FIRING
GFIRE2 B @FIRES2 GOTO FIRE SUBROUTINE

* PAUSE ROUTINE FOR THE TUNNEL SCENERIO. WILL RETURN TO BINC2 *
* UPON RETURN. *

PAUSE2 BL @PAUSES BRANCH TO PAUSE ROUTINE
 B @BINC2 AND NEXT KEY
PAGE

* THIS SUBROUTINE WILL SET THE VELOCITY TABLE TO MOVE LEFT OR RIGHT *

LR2 BL @CHKCRB CHECK FOR BOUNDARY CHARACTERS
 SWPB R7 YES WE CAN BECAUSE WE GOT BACK HERE!!
 BL @WRTVCL
 DATA SVT+>20,2,MYWS+14 AND PUT IT IN SVT FOR CREW MEMBER
 MOV B @JOYFLG,R3 DID WE USE JOYSTICK?

```

      JEQ LRRT2                NO, SO SCAN NEXT KEYBOARD
      B @CHKJB2              YES, SEE IF JOY WANTS TO FIRE
LRRT2  B @BINC2              CHECK NEXT KEYBOARD
*****
* THIS SUBROUTINE WILL SET THE VELOCITY TABLE TO MOVE UP OR DOWN *
*****
UD2    BL @CHKCRB            CHECK FOR BOUNDARY CHARACTERS FIRST
      BL @SEEUD              SEE IF WE CAN MOVE THAT WAY
      BL @WRTVCL
      DATA SVT+>20,2,MYWS+14  AND PUT IT IN SVT FOR CREW MEMBER
      MOVB @JOYFLG,R3        DID WE USE JOYSTICK?
      JEQ UDRT2              NO, SO SCAN NEXT KEYBOARD
      B @CHKJB2              YES, SEE IF JOY WANTS TO FIRE
UDRT2  B @BINC2              CHECK NEXT KEYBOARD
      PAGE
*****
* THIS NEXT ROUTINE DETERMINES IF THE CREW *
* MEMBER CAN MOVE IN THE SPECIFIED DIRECTION. *
* RETURN IS IN R12. *
*****
SEEUD  MOV R11,R12          SAVE RETURN
      BL @RDVCL             GET THE CREW MEMBER Y POSITION
      DATA SAL+>20,1,MYWS+2  Y OF TOP INTO R1 MSB
      SRL R1,8              INTO LSB OR R1
TOPBN2 CI R1,>0040          TOP BOUNDARY
      JLE CHKTD2            PAST IT, BUT CHECK DIR
BOTBN2 CI R1,>0067          RIGHT BOUNDARY
      JHE CHKBD2            PAST IT, BUT CHECK DIR
      JMP SEEUDR            IT'S O.K.
CHKTD2 MOV R7,R7           ARE WE MOVING UP?
      JGT SEEUDR            NO, SO DON'T STOP
      JMP STOPCR
CHKBD2 MOV R7,R7           ARE WE MOVING DOWN?
      JLT SEEUDR            NO, SO DON'T STOP
      JMP STOPCR
SEEUDR B *R12
      PAGE
*****
* STOP THE CREW MEMBER ROUTINE *
*****
STOPCR BL @WRTVCL          CREW SPRITE VELOCITY TABLE
      DATA SVT+>20,2,H0000  GO ZERO OUT THOSE TWO VELOCITIES
      MOVB @JOYFLG,R3        DID WE USE JOYSTICK?
      JEQ STCRRT            NO, SO SCAN NEXT KEYBOARD
      B @CHKJB2              YES, SEE IF JOY WANTS TO FIRE
STCRRT B @BINC2            AND CHECK NEXT KEY!!
      PAGE
*****
* THIS ROUTINE WILL DETERMINE THE CHARACER IN THE SIT *
* THAT THE CREW MEMBER IS ON. REMEMBER THAT THE Y PIXELS *
* START AT >FF, NOT >00 LIKE THE X'S. RETURN IS IN R13*
* --- SPECIAL NOTE --- UPON ENTRANCE, R6 CONTAINS *
* THE ADDRESS OF THE PARTICULAR BOUNDARY CHECK TABLE *
* FOR THE DIRECTION THE CREW MEMBER IS HEADING *
*****
CHKCRB MOV R11,R13          SAVE RETURN
      BL @RDVCL
      DATA SAL+>20,2,MYWS+2  Y, X INTO 1
*
      MOVB @CRWDIR,R8        GET DIRECTION WE'RE HEADING
      SRL R8,8              INTO LSB
      DEC R8                1.2.3.4 TO 0.1.2.3

```

```

CI R8,2          UP?
JNE CRBCNT      NO
CB @R1LB,@H60  WHAT SIDE ARE WE ON?
JL CRBCNT      LEFT, SO USE REGULAR UP INC VALUE
LI R8,4        RIGHT, SO USE SPECIAL UP INC VALUE
CRBCNT SLA R8,1  x BY 2 FOR DATA STATEMENTS
A @CBINCV(R8),R1 ADJUST THE SPRITE POSITION ACCORDINGLY
*
MOV R1,R2      AND MAKE A COPY
SRL R1,8      Y IN R1
INC R1        STARTS AT >FF
SRL R1,3      /Y BY 8
SLA R1,5      x BY 32 TO GIVE SIT ROW
ANDI R2,>00FF X IN LSB OF R2
SRL R2,3      /X BY 8
A R2,R1      SIT ADRESS OF WHERE SPRITE IS
*
MOV R1,VDPADD
BL @RDCL
DATA 1,MYWS+2 CHAR IN R1 MSBYTE
*
CBNDL MOVB *R6+,R5 GET THE FIRST CHECK CHARACTER
JEQ CBNDRT    DONE WITH LOOP, SO CAN MOVE
CB R1,R5     IS THIS A BOUNDARY CHAR?
JNE CBNDL    NO, SO KEEP CHECKING
B @STOPCR   YES, SO CANNOT MOVE ANYMORE
*
CBNDRT B *R13  LEGAL TO MOVE, SO RETURN TO CALLER
PAGE
*****
* THIS ROUTINE CHANGES THE PATTERN OF THE CREW MEMBER *
* RETURN LINKAGE IS IN R12. *
*****
CCPATT MOV R11,R12 SAVE RETURN
SB @H01,@CANDLY DECREMENT THE ANIM. DELAY COUNT
JNE CCPATR    NOT TIME TO CHANGE PATTERN YET
MOVB @H10,@CANDLY TIME TO CHANGE. RESET COUNTER
MOVB @CPCNTR,R5 NUMBER OF PATTERN CHANGES TO DATE
AB @CPINC,R5  TELLS WHICH PATTERN NEXT
MOVB R5,@CPCNTR AND SAVE THE COUNTER
BL @TSTCTR   TEST THE COUNTER FOR BOUNDARY
SRL R5,8     INTO LSB
SLA R5,5     x BY 32
A R5,@CGRMPT ADD OFFSET TO CURRENT GROM POINTER
MOV @CGRMPT,RLOC PREPARE TO GROM TO VDP
BL @GVCV
DATA 32,SDL+>140 GETS NEW ANIMATION PATTERN
MOVB @STRSD,R0 IS THE LASER FIRING?
JNE CCPATR   YES, SO DON'T USE STEP SOUND
BL @SOUND
DATA WLKSND EXECUTE MOVEMENT SOUND
CCPATR B *R12
PAGE
*****
* THIS SEES TO IT TO REVERSE THE PATTERN DIRECTION OF *
* THE CREW MEMBER. RETURN IS IN R13. *
*****
TSTCTR MOV R11,R13 SAVE RETURN
CB R5,@H03   PAST THE UPPER PATTERN BOUNDARY?
JEQ PATUP   YES, SO GO BACK DOWN
CB R5,@HFF   PAST THE LOWER PATTERN BOUNDARY?
JEQ PATDWN  YES, SO GO BACK UP

```



```

      B @TCTRRT
PATUP  MOV B @HFF,@CPINC      IT'S OK
      B @SETPFL              INCREMENT IS -1 TO GO DOWN
PATDWN MOV B @H01,@CPINC     AND SET THE PATTERN FLAG
SETPFL MOV B @H01,@CPCNTR   INCREMENT IS 1 TO GO UP
      MOV B @CPCNTR,R5      HAVE TO GO TO SECOND PATTERN
TCTRRT B *R13              AND PUT THAT INTO R5
      PAGE                  AND RETURN TO CALLER

*****
* THIS ROUTINE FIRES THE CREW MEMBER'S WEAPON *
*****
FIRES2 MOV B @CSHOTF,R3     IS BULLET STILL ON SCREEN?
      JNE F2RTRN           YES, SO CAN'T REFIRE
      MOV B @CLSLIM,R3    ARE WE OUT OF AMMO?
      JNE F2OK            NO, SO OK TO FIRE
      MOV B @H00,@CSHOTF  YES, SO CAN'T SEND A BULLET
      JMP F2G00N          BUT STILL CHECK FOR FIRE KEY
F2OK   MOV B @H01,@CSHOTF   NO, SO SET THE SHOT FLAG
*
F2G00N BL @COINC2          CHECK TO SEE IF WE'RE ON OBJECT
      DATA SAL+>28,>0404  OBJECT SPRITE AND Y,X TOLERANCES
      MOV R1,R1           ARE WE WALKING ON OBJECT?
      JEQ NOGOTW         NOT THIS SHOT
*
      MOV B @WONSCN,R0    ON SOMETHING... IS IT WATER?
      JEQ FONOBJ         NO
*
FONWAT BL @WRTVCL          TURN CREW MEMBER BLUE
      DATA SAL+>23,1,H04
      JMP DELOBJ
*
FONOBJ BL @RDVCL           GET COLOR OF SPRITE
      DATA SAL+>2B,1,MYWS+2
      BL @WRTVCL          AND TURN CREW MEMBER THAT COLOR
      DATA SAL+>23,1,MYWS+2
*
DELOBJ BL @WRTVCL          NO MORE WATER
      DATA SAL+>28,1,H00  NO MORE OBJECTS
      MOV B @H00,@WONSCN
      MOV B @H00,@OBJFLG
*
* AT THIS POINT HE IS ELIGIBLE TO RETURN TO THE SHIP
*
NOGOTW BL @GRMVDP         RESTORE TO FIRING AWAY POSITION
      DATA CREWU,32,SDL+>140
      MOV B @H00,@CRWDIR
*
      MOV B @CLSLIM,R3    ARE WE OUT OF AMMO
      JNE SCRBL          NO, SO SEND THE BULLET
      B @F2RTRN         YES, SO NO SHOT ALLOWED
*
SCRBL  LI R1,>9000        UP/DOWN BULLET PATTERN
      BL @WRTVCL
      DATA SAL+>12,1,MYWS+2
*
      BL @RDVCL          GET CREW Y, X
      DATA SAL+>20,2,MYWS+10  ADJUST FOR BULLET ARM
      AI R5,>F804        BULLET VELOCITY
      LI R7,>9C00
*
      BL @WRTVCL          PUT BULLET ON SCREEN
      DATA SAL+>10,2,MYWS+10

```

```

BL @WRTVCL
DATA SVT+>10, 2, MYWS+14      AND SET IT IN MOTION
*
SB @H01, @CLSLIM              ONE LESS SHOT
BL @SDUND                      EXECUTE FIRE SOUND
*
F2RTRN B @BINC2               AND CHECK NEXT KEYBOARD
PAGE
*****
* CHECK FOR BOUNDARY ON CREW BULLET.  RETURN *
* IS IN R15. *
*****
CSHOTB MOV R11, R15            SAVE RETURN
MOV B @CSHOTF, R1              IS SHOT ON SCREEN?
JEG CSBRTN                     NO, SO SKIP THIS MODULE
BL @RDVCL                      YES IT IS
DATA SAL+>10, 1, MYWS+2        GET BULLET Y INTO R1
SRL R1, 8                      PUT IN LSB
CI R1, 8                       PAST TOP BOUND?
JLE DELCBL                     YES, SO DELETE CREW BULLET
CI R1, >C1                     PAST BOTTOM OF SCREEN?
JHE DELCBL                     YES, SO DELETE
JMP CSBRTN                    IT'S O.K.
*
DELCBL BL @WRTVCL
DATA SAL+>10, 1, HCO          PUT BULLET OFF SCREEN
BL @WRTVCL
DATA SVT+>10, 2, H0000        AND ZERO VELOCITIES
MOV B @H00, @CSHOTF          SET THE FLAG TO NO SHOT
*
CSBRTN B *R15                 RETURN TO CALLER
PAGE
*****
* CHECK FOR COINC ON CREW MEMBER FROM MONSTER. *
* RETURN IS IN R13. *
*****
COINCC MOV R11, R13           SAVE RETURN
BL @COINC2
DATA SAL+>18, >OCOC          SPRITE AND TOLERANCE
MOV R1, R1                    DID MONSTER HIT CREW MEMBER?
JEG COCMN2                    NO, SO CHECK #2
CB @MN1FLG, @H05             IS MONSTER PAST SIZE 4?
JL COCMN2                    NO, SO CHECK MONSTER 2
JMP GOTCRW                   YES, SO CREW MEMBER BIT THE DUST
*
COCMN2 BL @COINC2
DATA SAL+>1C, >OCOC          MONSTER TWO AND TOLERANCE
MOV R1, R1                    DID MONSTER 2 HIT CREW?
JEG COCRTN                   NO, SO NO COINC HERE
CB @MN2FLG, @H05            IS MONSTER PAST SIZE 5?
JL COCRTN                   NO, SO NO COINC HERE
*
GOTCRW BL @WRTMSG
DATA 6                        COINC DETECTED ON CREW MEMBER!!
BL @SPSUB                    PUT MSG ON SCREEN
DATA 9                        MONSTER ATTACKED CREW
B @LSTCRW                    AND BRANCH TO LOST CREW ROUTINE
*
COCRTN B *R13                NO COINC, SO RETURN TO CALLER
PAGE
*****

```

* CHECK COINCIDENCE BETWEEN THE CREW MEMBER AND ANY OBJECT *
 * OR MONSTER. R1 SHOULD CONTAIN SAL OF MONSTER OR OBJECT AND*
 * R2 SHOULD CONTAIN Y, X TOLERANCE VALUES. THESE SHOULD BE *
 * PASSED IN DATA STATEMENTS. R1 WILL BE SET IF COINC IS DE- *
 * TECTED, ELSE IT WILL BE CLEAR. RETURN IS IN R12 *

COINC2 MOV #R11+,R1 SAL OF MONSTER OR OBJECT
 COI2YX MOV #R11+,R2 Y, X TOLS
 COI2 MOV R11,R12 SAVE RETURN
 *

BL @RDVCL R5=CREW Y, X
 DATA SAL+>20, 2, MYWS+10
 AI R5, >0808 ADJUST CREW MEMBER TO CENTER OF SPRITE
 MOV R1, VDPADD WHAT OBJECT TO CHECK
 BL @RDCL AND GET MONSTER OR OBJECT Y, X
 DATA 2, MYWS+14 *INTO R7
 CI R1, SAL+>28 ARE WE TESTING AN OBJECT?
 JEQ ADJOBV YES, SO ADJUST FOR AN OBJECT

*
 ADJMON AI R7, >0808 MOVE MONSTER POSITION TO CENTER
 JMP COI2CN AND CONTINUE
 ADJOBV AI R7, >FC08 MOVE WATER POSITION UP A BIT!!
 *

COI2CN MOV R5, R6 MAKE A COPY OF CREW POSITION
 SRL R5, 8 PUT Y INTO LSB OF R5
 ANDI R6, >00FF PUT X INTO LSB OF R6
 *

MOV R7, R8 MAKE A COPY OF MONS OR OBJ
 SRL R7, 8 PUT Y INTO LSB OF R7
 ANDI R8, >00FF PUT X INTO LSB OF R8

S R5, R7 GEY Y DIFFERENCE
 ABS R7 AND MAKE POSITIVE

S R6, R8 GEY X DIFFERENCE
 ABS R8 AND MAKE POSITIVE

CB @R7LB, R2 CHECK Y TOL
 JH NOCOI2 NO COINC HERE

CB @R8LB, @R2LB CHECK X TOL
 JH NOCOI2 NO COINC HERE

LI R1, 1 COINC DETECTED!!
 JMP COI2RT AND RETURN

*
 NOCOI2 CLR R1 NO COINC THIS TIME AROUND
 *

COI2RT B #R12 R1 CONTAIN CO/NO CO FLAG
 PAGE

 * RETURN CREW TO SHIP. NO RETURN IS NEEDED. *

RTCREW BL @GRMVDP RETURN CREW MEMBER TO STANDING POSITION
 DATA CREWU, 32, SDL+>140 AND STOP HIS MOTION
 BL @WRTVCL
 DATA SVT+>20, 2, H0000
 NOCBUL BL @WRTVCL ALSO STOP HIS BULLET IF HE JUST SHOT
 DATA SAL+>10, 1, HCO PUT BULLET OFF SCREEN
 BL @WRTVCL
 DATA SVT+>10, 2, H0000 AND ZERO VELOCITIES
 MOV B @H00, @CSHOTF SET THE FLAG TO NO SHOT

	BL @RDVCL	
	DATA SAL+>23, 1, MYWS+14	COLOR OF MAN INTO R7 MSB
	LI R1, >0600	RED/TRANS
	LI R5, 5	
*		
RCLP1	BL @WRTVCL	
	DATA SAL+>23, 1, MYWS+2	MAKE CREW MEMBER RED
	BL @ZDELAY	
	DATA >1800	WASTE TIME
	BL @WRTCL	
	DATA 1, R1LB	MAKE CREW MEMBER TRANS
	BL @ZDELAY	
	DATA >1800	WASTE TIME
	DEC R5	
	JNE RCLP1	NOT DONE WITH FLASHING
*		
	BL @WRTVCL	
	DATA SAL+>20, 1, HCO	OFF SCREEN FOR CREW MEMBER
	MOV B @H00, @CRWOUT	AND SET THE FLAG TO SAY SO
*		
	BL @SPSUB	
	DATA 18	
	BL @WRTMSG	
	DATA 15	CONGRATULATIONS
	BL @MSGDLY	AND WASTE TIME
*		
	CB R7, @H04	IS THIS GUY BLUE?
	JNE GOTBON	NO, SO DON'T FILL WATER GAGE
*		
	BL @SOUND	
	DATA GAGSND	WATER GAGE FULL SOUND
	BL @CLGAGE	FILL WATER GAGE
	BL @SPSUB	
	DATA 16	
	BL @WRTMSG	
	DATA 13	WATER TANKS FULL
	BL @MSGDLY	
GOTBON	BL @WRTMSG	
	DATA 2	BONUS POINTS GAINED
	BL @SPSUB	
	DATA 2	
	BL @MSGDLY	
	CB R7, @H04	DID HE GET WATER?
	JNE BONOBJ	NO, HE GOT AN OBJECT
BONWAT	LI R10, >2000	GOT WATER
	JMP REWRDS	
BONOBJ	LI R10, >1000	GOT AN OBJECT
*		
REWRDS	BL @SCORE	AND BONUS POINTS
	BL @SOUND	
	DATA BONSND	AND RING GONG!!
*		
	BL @MSGDLY	IN CASE EXTRA SHIP WAS AWARDED
	MOV B @H01, @LETMON	ALLOW MONSTERS NOW
	BL @GRMVPD	
	DATA BLANKM, 26, SIT+>1E3	BLANK OUT ANY MESSAGES
*		
	LI R5, 4	
	LI VDPADD, SAL+3	GUN COLOR BYTES
CLGUN2	BL @WRTCL	
	DATA 1, H06	RED
	AI VDPADD, 4	NEXT GUN

DEC R5
JNE CLGUN2

NOT DONE WITH LOOP

*
B @MAINLP
PAGE

RETURN CONTROL TO MAIN LOOP

* USER INTERRUPT ROUTINE FOR BOUNDARY CHECKS. *
* CONTROL PASSES HERE FROM INTERRUPT HANDLER *
* AND IS CURRENTLY IN GPL WORKSPACE. R13, *
* R14, AND R15 MUST BE PRESERVED SINCE THEY *
* ARE THE WS,PC, AND STATUS REG VALUES FOR *
* THE INTERRUPT WORKSPACE. *

* THIS MODULE CHECKS FOR BOUNDARIES AS THE MONSTERS ARE MOVING*
* AND REVERSES THEIR DIRECTION IF THEY HAVE PASSED THEM. *

CHKMON MOV B @H01, @INTFLG WE'RE NOW IN INTERRUPT ROUTINE
MOV B @MN1FLG, @MN1FLG MONSTER 1 ON SCREEN?
JEQ CHKMN2 NO, SO CHECK MONSTER 2
AB @H01, @M1CTIM INCREMENT MONSTER 1 TIMER BYTE
BL @RDVCL GO GET MONSTER POSITION
DATA SAL+>18, 2, GPLWS+12 INTO R6
BL @RDVCL GO GET MONSTER VELOCITIES
DATA SVT+>18, 2, GPLWS+16 INTO R8
MOV R6, R7 MAKE A COPY OF POSITION
MOV R8, R9 MAKE A COPY OF VELS
ANDI R6, >FF00 Y IN MSB OF R6
SLA R7, 8 X IN MSB OF R7
ANDI R8, >FF00 Y VEL IN MSB OF R8
SLA R9, 8 X VEL IN MSB OR R9
GTMBND LI R10, MBOUND ADDRESS OF MONSTER BOUNDARIES
CLR R12 CLEAR R12
MOV B @MN1FLG, R12 POINTER FOR OFFSET INTO MBOUND
SWPB R12 *INTO LSB
SLA R12, 2 AND MULTIPLY BY 4 (4 BOUNDARIES)
A R12, R10 AND ADD THE OFFSET
MOV B *R10+, R1 GET TOP BOUNDARY IN R1
MOV B *R10+, R2 GET BOT BOUNDARY IN R2
MOV B *R10+, R5 GET LFT BOUNDARY IN R5
MOV B *R10+, R12 GET RGT BOUNDARY IN R12

*
* NOTICE . . . NO FREE REGISTERS AT THIS POINT
*

CKMNLB CB R7, R5 IS MONSTER PAST LEFT BOUND?
JLE CKMNLD YES, BUT CHECK HIS DIRECTION
CKMNRB CB R7, R12 IS MONSTER PAST RIGHT BOUND?
JHE CKMNRD YES, BUT CHECK HIS DIRECTION
CKMNTB CB R6, R1 IS MONSTER PAST TOP BOUND?
JLE CKMNTD YES, BUT CHECK HIS DIRECTION
CKMNB B CB R6, R2 IS MONSTER PAST BOT BOUND?
JHE CKMNB D YES, BUT CHECK HIS DIRECTION
JMP CHKMN2 NO BOUNDARY VIOLATIONS!!!!

*
CKMNL D MOV B R9, R9 TEST AGAINST ZERO
JLT NEG X GOING LEFT, SO CHANGE DIR
JMP CKMNT B LETS GO TEST TOP OR BOT BOUND
CKMNR D MOV B R9, R9 TEST AGAINST ZERO
JGT NEG X GOING RIGHT, SO CHANGE DIR
JMP CKMNT B LETS GO TEST TOP OR BOT BOUND
CKMNT D MOV B R8, R8 TEST AGAINST ZERO
JLT NEG Y GOING UP, SO CHANGE DIR

CKMNB D JMP CHKBUL
MOV B R8, R8
JGT NEG Y
JMP CHKM N2

LETS GO CHECK BULLETS
TEST AGAINST ZERO
GOING DOWN, SO CHANGE DIR
LETS GO CHECK MONSTER 2

*
NEG X NEG R9
BL @CHM NVL
JMP CKM NTB
NEG Y NEG R8
BL @CHM NVL
JMP CHKM N2

MAKE X VEL OPPOSITE
AND REWRITE
NO GO CHECK UP DOWN BOUNDS
MAKE Y VEL OPPOSITE
AND REWRITE
NOW GO CHECK BULLETS

* SUBROUTINE TO CHANGE MONSTER'S VELOCITY. RETURN IS IN R10 *

CHM NVL MOV R11, R10
MOV B R9, @GR8LB
BL @WRTVCL
DATA SVT+>18, 2, GPLWS+16
B *R10

SAVE RETURN LINKAGE
REPLACE X VELOCITY IN R8
AND REWRITE TO SVT
*FROM R8 OF GPLWS
AND RETURN TO CALLER

*

PAGE

CHKM N2 MOV B @MN2FLG, @MN2FLG
JEG CHKBUL
AB @H01, @M2CTIM
BL @RDVCL
DATA SAL+>10, 2, GPLWS+12
BL @RDVCL
DATA SVT+>10, 2, GPLWS+16
MOV R4, R7
MOV R8, R9
ANDI R6, >FF00
SLA R7, 8
ANDI R8, >FF00
SLA R9, 8
GT2BND LI R10, MBOUND
CLR R12
MOV B @MN2FLG, R12
SWPB R12
SLA R12, 2
A R12, R10
MOV B *R10+, R1
MOV B *R10+, R2
MOV B *R10+, R5
MOV B *R10+, R12

ARE MONSTERS ON SCREEN?
NO, SO CHECK BULLETS
INCREMENT MONSTER 2 TIMER BYTE
GO GET MONSTER POSITION
INTO R6
GO GET MONSTER VELOCITIES
INTO R8
MAKE A COPY OF POSITION
MAKE A COPY OF VELS
Y IN MSB OF R6
X IN MSB OF R7
Y VEL IN MSB OF R8
X VEL IN MSB OR R9
ADDRESS OF MONSTER BOUNDARIES
CLEAR R12
POINTER FOR OFFSET INTO MBOUND
INTO LSB
AND MULTIPLY BY 4 (4 BOUNDARIES)
AND ADD THE OFFSET
GET TOP BOUNDARY IN R1
GET BOT BOUNDARY IN R2
GET LFT BOUNDARY IN R5
GET RGT BOUNDARY IN R12

*

* NOTICE . . . NO FREE REGISTERS AT THIS POINT

*

CKM2LB CB R7, R5
JLE CKM2LD
CKM2RB CB R7, R12
JHE CKM2RD
CKM2TB CB R6, R1
JLE CKM2TD
CKM2BB CB R6, R2
JHE CKM2BD
JMP CHKBUL

IS MONSTER PAST LEFT BOUND?
YES, BUT CHECK HIS DIRECTION
IS MONSTER PAST RIGHT BOUND?
YES, BUT CHECK HIS DIRECTION
IS MONSTER PAST TOP BOUND?
YES, BUT CHECK HIS DIRECTION
IS MONSTER PAST BOT BOUND?
YES, BUT CHECK HIS DIRECTION
NO BOUNDARY VIOLATIONS!!!!

*

CKM2LD MOV B R9, R9
JLT NEG X2
JMP CKM2TB
CKM2RD MOV B R9, R9
JGT NEG X2

TEST AGAINST ZERO
GOING LEFT, SO CHANGE DIR
LETS GO TEST TOP OR BOT BOUND
TEST AGAINST ZERO
GOING RIGHT, SO CHANGE DIR

```

      JMP CKM2TB          LETS GO TEST TOP OR BOT BOUND
CKM2TD MOVB R8,R8        TEST AGAINST ZERO
      JLT NEGY2          GOING UP , SO CHANGE DIR
      JMP CHKBUL        LETS GO CHECK BULLETS
CKM2BD MOVB R8,R8        TEST AGAINST ZERO
      JGT NEGY2          GOING DOWN, SO CHANGE DIR
      JMP CHKBUL        LETS GO CHECK BULLETS
*
NEGX2  NEG R9            MAKE X VEL OPPOSITE
      BL @CHM2VL        AND REWRITE
      JMP CKM2TB        NO GO CHECK UP DOWN BOUNDS
NEGY2  NEG R8            MAKE Y VEL OPPOSITE
      BL @CHM2VL        AND REWRITE
      JMP CHKBUL        NOW GO CHECK BULLETS
*****
* SUBROUTINE TO CHANGE MONSTER'S VELOCITY. RETURN IS IN R10 *
*****
CHM2VL MOV R11,R10      SAVE RETURN LINKAGE
      MOVB R9,@GR5LB    REPLACE X VELOCITY IN R8
      BL @WRTVCL        AND REWRITE TO SVT
      DATA SVT+>1C,2,GPLWS+16 *FROM R8 OF GPLWS
      B *R10            AND RETURN TO CALLER
*
      PAGE
*****
* THIS MODULE CHECKS FOR BOUNDARIES AS THE BULLETS ARE MOVING *
* AND DELETES THEM IF THEY HAVE PASSED THE BOUNDARIES. *
*****
CHKBUL MOVB @SHOTFL,R0  ARE BULLETS ON SCREEN?
      JEQ STPBUL        NO, SO DON'T BOTHER WITH BOUNDARIES
*
* THE ABOVE LINE WAS CHANGED FROM 'JEQ EVENTB' TO 'JEQ STPBUL'
*
      BL @RDVCL
      DATA SAL+16,6,GPLWS+10  2 BULLETS Y,X INTO R5,R7
      SRL R6,8                AND PATTERN VALUE TO LSB OR R6
      CI R6,>0090             ARE BULLETS VERTICAL?
      JEQ UDBULS             YES, SO CHECK VERTICAL BOUNDARIES
LRBULS LI R10,>0058          CENTER OF SCREEN BOUNDARY
      CB @GR5LB,@GRALB      X OF SPRITE PAST BOUNDARY?
      JHE STPBUL            YES, SO STOP AND DELETE
      LI R10,>00A8
      CB @GR7LB,@GRALB      X OF SPRITE PAST BOUNDARY?
      JLE STPBUL            YES, SO STOP AND DELETE
      JMP SPKINT            NO, SO DON'T DELETE BULLET, YET
UDBULS LI R10,>4400          CENTER OF MINE BOUNDARY
      CB R5,R10             Y OF SPRITE PAST BOUNDARY?
      JHE STPBUL            YES, SO DELETE BULLETS
      LI R10,>3C00
      CB R7,R10             Y OF SPRITE PAST BOUNDARY?
      JLE STPBUL            YES, SO DELETE BULLETS
      JMP SPKINT            NO, SO DON'T DELETE BULLET, YET
STPBUL MOVB @CRWOUT,R3     WAS IT A CREW SHOT?
      JNE SPKINT            YES, SO DON'T STOP THE BULLETS HERE
      BL @WRTVCL
      DATA SVT+16,2,H0000    PUT ZERO VELOCITIES IN THE BULLETS
      BL @WRTVCL
      DATA SVT+20,2,H0000    SAME HERE
DELBUL BL @WRTVCL
      DATA SAL+16,1,HCO
      BL @WRTVCL
      DATA SAL+20,1,HCO

```

MOV B @H00, @SHOTFL

TURN OFF SHOT FLAG!

PAGE

*** 9900 CODE INTERRUPT DRIVEN SPEECH WITH GPL SET UP AND EXECUTION ***

* USER INTERRUPT ROUTINE FOR SPEECH. *
* CONTROL PASSES HERE FROM INTERRUPT HANDLER *
* AND IS CURRENTLY IN GPL WORKSPACE. R13, *
* R14, AND R15 MUST BE PRESERVED SINCE THEY *
* ARE THE WS, PC, AND STATUS REG VALUES FOR *
* THE INTERRUPT WORKSPACE. *

*
* VARIABLE DEFINITION FOR INTERRUPT-DRIVEN
* SPEECH PROCESSING. INITIAL DESIGN IS FOR
* LOADING DIRECT SPEECH DATA.
*

ADDR	DEFINITION
SPADR	CURRENT POINTER TO LPC SPEECH DATA
SPLN	NO. OF BYTES REMAINING IN LIST

*
* THE POINTER @SPADR IS CLEARED AFTER THE SPEECH
* DATA HAS FED TO THE SPEECH CHIP
*

*
* UPON ENTRY TO INTERRUPT ROUTINE, THE WORD @SPADR
* IS COMPARED TO ZERO. IF EQUAL, THIS INDICATES
* NO SPEECH DATA AVAILABLE, AND INTERRUPT PROCESSING
* IS TERMINATED. IF NONZERO, A CHECK IS DONE FOR
* START OF NEW SPEECH LIST. THIS IS INDICATED BY
* A VALUE OF 1-255 @SPADR. VALUES IN THIS RANGE WILL
* WILL BE CONSIDERED INDEXES INTO AN LPC ADDRESS
* POINTER TABLE. EACH ADDRESS REPRESENTS THE START
* OF LPC DATA FOR THE SPECIFIED PHRASE.
*

* REGISTER USAGE

* R0 - R5 HOLDS 9900 CODE WHICH READS SPEECH STATUS
* R6 USER GROM ADDRESS SAVE REGISTER
* R7 TEMPORARY
* R8 ADDRESS POINTER TO LPC DATA
* R9 TEMPORARY
* R10 DESTINATION REGISTER FOR SPEECH STATUS INFO
*

SPCHRD EQU >9000
SPCHWT EQU SPCHRD+>400
READIT EQU GPLWS
SPDATA EQU GPLWS+20 LOCATION IS GPL WORKSPACE REG 10
*
SPKINT EQU \$


```

MOV @SPADR, R8      AFFECT STATUS W/ADDRESS POINTR
JEQ SPKRT2          IF NO POINTER, EXIT
LI R7, CODADR       ADDRESS OF CODE TO COPY
LI R10, READIT      LOAD POINT IN FAST RAM
LI R9, CDLEN        LENGTH OF CODE BLOCK
MOVEIT MOV *R7+, *R10+  MOV ONE WORD
DECT R9             DECREMENT WORD COUNTER
JNE MOVEIT          MOVE MORE IF NOT FINISHED
LI R9, 8            DEFAULT NO. OF BYTES TO MOVE
MOV B R8, R7        CHECK FOR START OF NEW PAGE
JNE GRMFX           ADDRESS SET UP, SO MOVE DATA
BL @READIT          GET SPEECH STATUS
SLA R10, 1          CHECK 'SPEECH DONE?' BIT
JOC SPKRT2          IF STILL BUSY, RETURN
MOV R8, R10         COPY TABLE INDEX VALUE
SLA R10, 1          MAKE VALUE A WORD INDEX

*===>
*===> IF SPANISH IS USED, SWITCH THE TABLES HERE!!
*===>

MOV B @WHICHL, @WHICHL  ARE WE USING SPANISH?
JEQ USEENG             NO, SO USE ENGLISH
USES PA MOV @ESDXTB(R10), R10  COPY GROM ADDRESS FROM TABLE
JMP USELNG             AND CONTINUE ON
USEENG MOV @INDXTB(R10), R10  COPY GROM ADDRESS FROM TABLE
USELNG BL @SAVIT2          SAVE OLD GROM ADDRESS
MOV R10, R8            COPY GROM ADR TO ADR DECR REG
BL @GRMAD2             SET UP GROM ADDRESS
MOV B @GRMRD, @SPLEN  GET FIRST BYTE OF LENGTH
MOV B @GRMRD, @SPLEN+1  GET SECOND LENGTH BYTE
INCT R8                ADJUST GROM POINTER TO ACTUAL DATA
* MOV R8, @SPADR        STORE ADDR IN PERMANENT LOC
* BL @SPCMND            SET UP 'SPEAK EXTERNAL' COMMAND
SPCMND

MOV B @H60, @SPCHWT   SEND THE 'SPEAK EXTERNAL' CMND
NOP
NOP
NOP
NOP
NOP
* RT
LI R9, 16              INIT BYTE COUNT TO FILL SPEECH FIFO
JMP SPCHST            CHECK SPEECH STATUS
GRMFX EQU $
LI R9, 8               8 BYTES TO SEND TO SPEECH FIFO
BL @SAVIT2             SAVE OLD GROM ADDRESS
SPCHST EQU $
BL @READIT             READ SPEECH CHIP STATUS
SLA R10, 2             MOV 2ND MSBIT INTO CARRY BIT
JNC SPKRET            IF BUFFER NOT LOW, JUMP
MOV R8, R10            GET READY TO PASS GROM ADDRESS
BL @GRMAD2             SET UP THE GROM ADDRESS
*
*
LOOPR EQU $
MOV B @GRMRD, @SPCHWT  MOVE ONE BYTE TO SPEECH FIFO
DEC @SPLEN             DECREMENT TOTAL BYTE COUNTER
JEQ SPKRET             IF ZERO, JUMP TO EXIT
INC R8                 POINT TO NEXT SPEECH BYTE IN GROM
DEC R9                 DECREMENT LOCAL FIFO FILL CNT
JNE LOOPR              JUMP AND MOVE NXT BYTE IF >0
CLR @SPADR             FIFO ZERO, SO DON'T EXECUTE PHRASE ANYMORE
JMP SPKRT3            RESTORE OLD GROM ADR!!!!!!!!!!!!
*

```

```

*
SPKRET EQU $
MOV @SPLEN, R7 CHECK FOR ZERO VALUE IN LENGTH
JNE SPKRT3 IF LEN NOT ZERO, RESTORE PNTR
CLR R8 CLEAR POINTER VAL; USE AS FLAG
SPKRT3 EQU $
MOV R6, R10 GET USER GROM COUNTER VALUE
BL @GRMAD2 RESTORE USER GROM ADDRESS

*
SPKRT2 EQU $
MOV R8, @SPADR RESTORE CURRENT POINTER VALLUE

*
MOVB @H00, @INTFLG NO LONGER IN INTERRUPTS
LWPI INTWS RESTORE INTERRUPT LINKAGE
RTWP RETURN TO CALLING ASMBLY PRGRM
PAGE

*****
*** SHORT SUBROUTINE TO SAVE GROM ADDRESS FOR MAIN PROGRAM *
*****
SAVIT2 EQU $
MOVB @GRMRA, R6 SAVE ONE BYTE OF GROM ADDRESS
MOVB @GRMRA, @GPLWS+13 SAVE LOW BYTE OF GPL ADDRESS
DEC R6 ADJUST GROM ADDRESS FOR NEXT SETUP
RT

*****
*** DELAY LOOP FOR 16 BIT BUSS *
*****
CODADR MOVB @SPCHR, R10
NOP
NOP
NOP
RT
CDLEN EQU $-CODADR
*****
* ROUTINE TO SET UP GROM *
* READ ADDRESS WINDOW *
*****
GRMAD2 EQU $
MOVB R10, @GRMWA
SWPB R10
MOVB R10, @GRMWA
RT
PAGE

*****
*** GROM ADDRESSES REPRESENTING START OF LPC SPEECH DATA FOR ENG PHRASES *
*****
SPADDR EQU >A000 GROM 5 . . . START OF SPEECH DATA
INDXTB EQU $-2 REMEMBER THAT A ZERO INDEX
* VALUE WILL NEVER OCCUR(HA! HA!)
SPADVA DATA SPADDR 1 ADVANCE TO NEW LEVEL
ACCUM1 EQU SPADDR+2+270
SPBONU DATA ACCUM1 2 BONUS POINTS GAINED
ACCUM2 EQU ACCUM1+2+222
SPEXTR DATA ACCUM2 3 EXTRA CREW MEMBER
ACCUM3 EQU ACCUM2+2+200
SPGOOD DATA ACCUM3 4 GOOD SHOT, CAPTAIN
ACCUM4 EQU ACCUM3+2+218
SPMDES DATA ACCUM4 5 MONSTER DESTROYED
ACCUM5 EQU ACCUM4+2+210
SPWAY DATA ACCUM5 6 WAY TO GO, CAPTAIN
ACCUM6 EQU ACCUM5+2+222
SPLASR DATA ACCUM6 7 LASERS OVERHEATED

```

ACCUM7 EQU ACCUM6+2+290
 SPMDAM DATA ACCUM7
 ACCUMB EQU ACCUM7+2+234
 SPMATT DATA ACCUMB
 ACCUM9 EQU ACCUMB+2+220
 SPCREW DATA ACCUM9
 ACCUMA EQU ACCUM9+2+154
 SPDETC DATA ACCUMA
 ACCUMB EQU ACCUMA+2+288
 SPZYGO DATA ACCUMB
 ACCUMC EQU ACCUMB+2+218
 SPCONT DATA ACCUMC
 ACCUMD EQU ACCUMC+2+238
 SPCOOL DATA ACCUMD
 ACCUME EQU ACCUMD+2+224
 SPWATR DATA ACCUME
 ACCUMF EQU ACCUME+2+196
 SPFULL DATA ACCUMF
 ACCUMG EQU ACCUMF+2+194
 SPOUT DATA ACCUMG
 ACCUMH EQU ACCUMG+2+214
 SPCONG DATA ACCUMH
 ACCUMI EQU ACCUMH+2+264
 SPLAFS DATA ACCUMI
 ACCUMJ EQU ACCUMI+2+114
 SPTHRT DATA ACCUMJ
 ACCUMK EQU ACCUMJ+2+180
 PAGE

8 MONSTER DAMAGED SHIP
 9 MONSTER ATTACKED CREW
 10 CREW MEMBER LOST
 11 DETECTING UNKNOWN OBJECT
 12 ZYGONAUT APPROACHING
 13 CONTINUE GAME, CAPTAIN
 14 COOLANT IS RUNNING LOW
 15 WATER AHEAD
 16 WATER TANKS FULL
 17 OUT OF WATER, CAPTAIN
 18 CONGRATULATIONS, CAPTAIN
 19 HA HA HA HA
 20 YOU'LL NEVER GET ME

 *** GROM ADDRESSES REPRESENTING START OF LPC SPEECH DATA FOR SPA PHRASES *

ESADDR EQU >C000
 ESDXTB EQU \$-2
 *

GROM 6 . . . START OF SPEECH DATA

ESADVA DATA ESADDR
 ECCUM1 EQU ESADDR+2+274
 ESBONU DATA ECCUM1
 ECCUM2 EQU ECCUM1+2+318
 ESEXTR DATA ECCUM2
 ECCUM3 EQU ECCUM2+2+344
 ESGOOD DATA ECCUM3
 ECCUM4 EQU ECCUM3+2+214
 ESMDES DATA ECCUM4
 ECCUM5 EQU ECCUM4+2+242
 ESWAY DATA ECCUM5
 ECCUM6 EQU ECCUM5+2+234
 ESLASR DATA ECCUM6
 ECCUM7 EQU ECCUM6+2+278
 ESM DAM DATA ECCUM7
 ECCUMB EQU ECCUM7+2+344
 ESMATT DATA ECCUMB
 ECCUM9 EQU ECCUMB+2+442
 ESCREW DATA ECCUM9
 ECCUMA EQU ECCUM9+2+346
 ESDETC DATA ECCUMA
 ECCUMB EQU ECCUMA+2+362
 ESZYGO DATA ECCUMB
 ECCUMC EQU ECCUMB+2+276
 ESCONT DATA ECCUMC
 ECCUMD EQU ECCUMC+2+324
 ESCOOL DATA ECCUMD
 ECCUME EQU ECCUMD+2+282

1 ADVANCE TO NEW LEVEL
 2 BONUS POINTS GAINED
 3 EXTRA CREW MEMBER
 4 GOOD SHOT, CAPTAIN
 5 MONSTER DESTROYED
 6 WAY TO GO, CAPTAIN
 7 LASERS OVERHEATED
 8 MONSTER DAMAGED SHIP
 9 MONSTER ATTACKED CREW
 10 CREW MEMBER LOST
 11 DETECTING UNKNOWN OBJECT
 12 ZYGONAUT APPROACHING
 13 CONTINUE GAME, CAPTAIN
 14 COOLANT IS RUNNING LOW

```

ESWATR DATA ECCUME
ECCUMF EQU ECCUME+2+178
ESFULL DATA ECCUMF
ECCUMG EQU ECCUMF+2+360
ESOUT DATA ECCUMG
ECCUMH EQU ECCUMG+2+264
ESCONG DATA ECCUMH
ECCUMI EQU ECCUMH+2+282
ESLAFS DATA ECCUMI
ECCUMJ EQU ECCUMI+2+114
ESTHRT DATA ECCUMJ
ECCUMK EQU ECCUMJ+2+304
PAGE

```

```

15 WATER AHEAD
16 WATER TANKS FULL
17 OUT OF WATER, CAPTAIN
18 CONGRATULATIONS, CAPTAIN
19 HA HA HA HA
20 YOU'LL NEVER GET ME

```

```

*****
* STANDARD KEYBOARD SCAN RTN*
*****
SCANKY LIM1 0

```

```

LWPI GPLWS
BL @SCAN
LWPI MYWS
LIMI 2
B *R11

```

```

*****
* SAVE GPC AND CLEAR SPEECH *
*****

```

```

SAVGPL MOVB @GRMRA,@GPCSAV
MOV B @GRMRA,@GPCSAV+1
DEC @GPCSAV
CLR @SPADR
CLR @SPLEN
B *R11

```

SAVE GPL PROGRAM COUNTER

GRMRA ACCESS AUTO INCREMENTING
NO SPEECH EXECUTING
SO NONE TO COUNT DOWN
AND RETURN

```

*****
* CALL SOUND ROUTINE *
*****

```

```

SOUND MOV *R11+,SNDPNT
LIMI 0
SOCB @H01,@RAMFLG
MOV B @H01,@STRTSD
LIMI 2
B *R11
PAGE

```

LOAD DATA POINTER

SOUND LIST IS IN VDRAM
START THE SOUND PROCESSING

```

*****
* SET UP SPEECH ROUTINE *
*****

```

```

SPSUB MOV @SPADR,@SPADR
JNE NOSPCH
MOV *R11,@SPADR
NOSPCH INCT R11
B *R11

```

IS SPEECH STILL EXECUTING
YES, SO NO SPEECH TO SEND
SET THE INDEX FOR SPEECH PHRASE
POINT BACK TO NEXT STATEMENT

```

*****
* WRITE MSG SUBROUTINE *
*****

```

```

WRTMSG MOV *R11+,R3
MOV R11,@>B37E
DEC R3
CLR RLOC
LI R0,26
MPY R0,R3
A @LANGPT,RLOC
BL @GVCV
DATA 26,SIT+>1E3
MOV @>B37E,R11

```

WHERE'S THE DATA TO WRITE
TEMPORARY RETURN NOT USED BY ANYONE
ACTUALLY STARTS FROM 0
PREPARE FOR MULTIPLY
MULTIPLICAND
THIS CREATES OFFSET IN RLOC (R4)
NOW INTO WHICH TABLE?

WRITE IT OUT
RESTORE RETURN

```

        B *R11
        PAGE
*****
* READ FROM VDP *
*****
RDVCL  MOV *R11+,VDPADD
RDCL   MOV *R11+,RCOUNT
RDL    MOV *R11+,RLOC
READ   LIM 0
        SWPB VDPADD
        MOVB VDPADD,@VDPWA
        SWPB VDPADD
        MOVB VDPADD,@VDPWA
        NOP
R00010 MOVB @VDPRD,*RLOC+
        DEC RCOUNT
        JGT R00010
        MOVB @INTFLG,@INTFLG
        JNE RRT
        LIM 2
RRT    RT
        PAGE
*****
* WRITE TO VDP *
*****
WRTVCL MOV *R11+,VDPADD
WRTCL  MOV *R11+,WCOUNT
WRTL   MOV *R11+,WLOC
WRITE  LIM 0
        SWPB VDPADD
        MOVB VDPADD,@VDPWA
        SWPB VDPADD
        ORI  VDPADD,>4000
        MOVB VDPADD,@VDPWA
        NOP
W00010 MOVB *WLOC+,@VDPWD
        DEC WCOUNT
        JGT W00010
        LI  WCOUNT,>4000
        S  WCOUNT,VDPADD
        MOVB @INTFLG,@INTFLG
        JNE WRT
        LIM 2
WRT    RT
*****
* WRITE TO REG *
*****
***>>> VALUE TO BE PASSED IS IN THIS FORMAT: VALUE!B!REG #
VWTR   MOV *R11+,WLOC
VWTRW  LIM 0
        MOVB WLOC,@VDPWA
        SWPB WLOC
        MOVB WLOC,@VDPWA
        SWPB WLOC
        LIM 2
        RT
        PAGE
*****
* WRITE FROM GROM TO VDP*
*****
*****
* THIS ROUTINE ASSUMES THAT THE PARAMETERS ARE IN A DATA STATEMENT IM- *

```

AND RETURN

DISABLE INTERRUPTS

SET UP LOWER BYTE OF ADDRESS

SET UP UPPER BYTE OF ADDRESS

READ VDP BYTE INTO BUFFER

DECREMENT COUNTER

AND LOOP IF NEEDED

ARE WE IN INTERRUPTS?

YES, SO DON'T TURN THEM ON

ENABLE INTERRUPTS

DISABLE INTERRUPTS

SET UP LOWER BYTE OF ADDRESS

SET VDP WRITE FLAG

SET UP UPPER BYTE OF ADDRESS

WRITE BYTE INTO VDP

DECREMENT COUNTER

THIS IS WHAT WE OR'D

RESTORE VDPADD

ARE WE IN AN INTERRUPT?

YES, SO DON'T TURN THEM ON

ENABLE INTERRUPTS

SEND LOW BYTE

WASTE TIME

SEND HI BYTE

WASTE TIME

```

* MEDIATELY FOLLOWING THE BL @GRMVDP. FOR EXAMPLE,
* BL @GRMVDP
* DATA MONSTA, 32, SDT+>64
*****
GRMVDP MOV *R11+, RLOC FROM WHERE IN GROM
GVCV MOV *R11+, RCOUNT HOW MANY BYTES
GVV MOV *R11+, VDPADD TO WHERE IN VDP
GV LIM 0 DISABLE INTERRUPTS
MOV RLOC, @GRMWA SET UP UPPER BYTE OF GROM ADDR
SWPB RLOC
MOV RLOC, @GRMWA SET UP LOWER BYTE OF GROM ADDR
MOV @VADDLB, @VDPWA SET UP LOWER BYTE FOR VDP
ORI VDPADD, >4000 SET VDP WRITE FLAG
MOV VDPADD, @VDPWA SET UP UPPER BYTE OF ADDRESS
NOP
W00030 MOV @GRMRD, @VDPWD WRITE BYTE INTO VDP
DEC RCOUNT DECREMENT COUNTER
JGT W00030
LI RCOUNT, >4000 RESTORE VDPADD
S RCOUNT, VDPADD
LIM 2 ENABLE INTERRUPTS
RT
PAGE
*****
* GROM READ SUBROUTINES FOR WALL INCREMENTING
*****
GSBR2 LIM 0
MOV R3, @GRMWA * SET MSB
SWPB R3 *
MOV R3, @GRMWA * SET LSB
NOP
MOV @GRMRD, R3 * GET ADDRESS
SWPB R3
MOV @GRMRD, R3
SWPB R3
LIM 2
B *R11
GSBR1 LIM 0
MOV R3, @GRMWA * SET MSB
SWPB R3 *
MOV R3, @GRMWA * SET LSB
NOP
MOV @GRMRD, R3 * GET ADDRESS
SWPB R3
MOV @GRMRD, R3
SWPB R3
MOV R3, @GRMWA * SET MSB
SWPB R3
MOV R3, @GRMWA * SET LSB
LIM 2
B *R11
*
LIST
END

```

TITLE MOONMINE
GROM 3
ORG 0

```
*****  
*           M O O N   M I N E           *  
*                                           *  
*           BY : JOHN PHILLIPS           *  
*           DATE: 01/03/83               *  
*           FILE: .JMP089.PROC.MOONG     *  
*                                           *
```

```
*****  
*****  
*                                           *
```

```
*           G R O M   A L L O C A T I O N           *  
*                                           *  
* GROM 3  GPL CODE, LANGUAGE PHRASES     *  
* GROM 4  MONSTERS, WATER, CREW PATTERNS, SOUND LISTS *  
*           GENERAL DATA                 *  
* GROM 5  ENGLISH SPEECH PHRASES         *  
* GROM 6  SPANISH SPEECH PHRASES         *  
* GROM 7  SCREEN LOAD DATA, TITLE SCREEN CHARACTERS, *  
*           TITLE SCREEN SIT FORMAT, BOT 1/3 PLAYING AREA *  
*                                           *
```

```
*****  
*****  
* GROM 4  EQUATES FOR PARAMETER PASSING *  
*****
```

```
VDPNOR EQU >8000  
VDPOFF EQU >8002  
COLORS EQU >8004  
TITCOL EQU >8006  
CPYRIT EQU >8008  
BLANKS EQU >800A  
GBINIT EQU >800C  
ULCORN EQU >800E  
BLANK EQU >8010  
CREWP EQU >8012  
LGAGE EQU >8014  
AURL EQU >8016  
TGAGE1 EQU >8018  
GUNS EQU >801A  
FIRSND EQU >801C  
GAGSND EQU >801E
```

```
*****  
* GROM 5  EQUATES FOR PARAMETER PASSING *  
*****
```

```
TITCHR EQU >E000  
TITSIT EQU >E002
```

```
*****  
* CPU EQUATES *  
*****
```

```
R0 EQU >00  
ROLB EQU >01  
R1 EQU >02  
R1LB EQU >03  
R2 EQU >04  
R2LB EQU >05  
CB/ADD EQU >06  
R3 EQU >06  
R3LB EQU >07  
FMT EQU >08  
R4 EQU >08  
FMT1 EQU >09
```

```

R4LB EQU >09
LENGTH EQU >0A
R5 EQU >0A
R5LB EQU >0B
CNTFMT EQU >0C
R6 EQU >0C
R6LB EQU >0D
R7 EQU >0E
R7LB EQU >0F
R8 EQU >10
R8LB EQU >11
R9 EQU >12
R9LB EQU >13
R10 EQU >14
R10LB EQU >15
R11 EQU >16
R11LB EQU >17
R12 EQU >18
R12LB EQU >19
R13 EQU >1A
R13LB EQU >1B
R14 EQU >1C
R14LB EQU >1D
R15 EQU >1E
R15LB EQU >1F
CHEATF EQU >23
WHICHL EQU >3A
TANKS EQU >4B
GAMELV EQU >49
LANGPT EQU >6A
LANG EQU >6C
DEMOFL EQU >6E
KEYBRD EQU >74
KEY EQU >75
JOYY EQU >76
JOYX EQU >77
TIMER EQU >79
KEEPR1 EQU >D4
CLRSCN EQU >D6
THRU EQU >FF

```

```

CHEAT FLAG 0-NO CHEAT
0-ENGLISH ?-SPANISH

```

```

FOR TEXT FORMATTER ROM
FOR TEXT FORMATTER GROM
DEMO FLAG 0=NO DEMO 1=DEMO
KEY BOARD TO SCAN FROM
KEY CODE RETURNED

```

```

TIMER?
TO KEEP VDP R1
SCREEN TIME OUT COUNTER

```

*

```

*****
* MEMORY MAPPED EQUATES *

```

```

*****
PDT EQU >0800 PATTERN DESCR TABLE
SIT EQU >0000 SCREEN IMAGE TABLE
SAL EQU >0300
CT EQU >0380 COLOR TABLE
SDL EQU >0400

```

```

*****
* TI SUBROUTINE EQUATES *

```

```

*****
FAC EQU >4A 2 BYTES, POINTER TO SMALL CHARS SET

```

```

*****
* TEXT FORMATTER EQUATES *

```

```

*****
* TEMP. 0
* WATER 1
* LEVEL 2
* CREW: 3
* SCORE: 4
* HIGH : 5

```


* GAME 6
* OVER 7
* PRESS: 8
* REDO OR BACK 9

* GROM Header *

GHEADR DATA >AA, 1, 1, 0, #0, #PROG1, #0, #0, #0
PROG1 DATA #PROG2, #STSWD, 7, : SVENSKA:
PROG2 DATA #PROG3, #STDUT, 10, : NEDERLANDS:
PROG3 DATA #PROG4, #STSPA, 7, : ESPANOL:
PROG4 DATA #PROG5, #STITA, 8, : ITALIANO:
PROG5 DATA #PROG6, #STGER, 7, : DEUTSCH:
PROG6 DATA #PROG7, #STFRE, 8, : FRANCAIS:
PROG7 DATA #0, #STENG, 7, : ENGLISH:

* SET PROPER FLAGS FOR LANGUAGE POINTERS *

STENG DST ENGPHR, @LANGPT FOR ROM CODE
DST CONENG, @LANG FOR TEXT FORMATTER
ST 0, @WHICHL USING ENGLISH, FOLKS
B STARTG
STSWD DST SWDPHR, @LANGPT
DST CONSWD, @LANG FOR TEXT FORMATTER
ST 0, @WHICHL USING ENGLISH, FOLKS
B STARTG
STDUT DST DUTPHR, @LANGPT
DST CONDUT, @LANG FOR TEXT FORMATTER
ST 0, @WHICHL USING ENGLISH, FOLKS
B STARTG
STSPA DST SPAPHR, @LANGPT
DST CONSPA, @LANG FOR TEXT FORMATTER
ST 1, @WHICHL USING SPANISH, FOLKS
B STARTG
STITA DST ITAPHR, @LANGPT
DST CONITA, @LANG FOR TEXT FORMATTER
ST 0, @WHICHL USING ENGLISH, FOLKS
B STARTG
STGER DST GERPHR, @LANGPT
DST CONGER, @LANG FOR TEXT FORMATTER
ST 0, @WHICHL USING ENGLISH, FOLKS
B STARTG
STFRE DST FREPHR, @LANGPT
DST CONFRE, @LANG FOR TEXT FORMATTER
ST 0, @WHICHL USING ENGLISH, FOLKS
B STARTG

* CONSOLE TEXT STATEMENTS *

CONENG DATA #CENG1
DATA #CENG2
DATA #CENG3
DATA #CENG4
DATA #CENG5
DATA #CENG6
DATA #CENG7
DATA #CENG8
DATA #CENG9
DATA #CENG10
DATA #ENGEOL

*
CONSWD DATA #CSWD1

DATA #CSWD2
DATA #CSWD3
DATA #CSWD4
DATA #CSWD5
DATA #CSWD6
DATA #CSWD7
DATA #CSWD8
DATA #CSWD9
DATA #CSWD10
DATA #SWDEOL

*

CONDUT DATA #CDUT1
DATA #CDUT2
DATA #CDUT3
DATA #CDUT4
DATA #CDUT5
DATA #CDUT6
DATA #CDUT7
DATA #CDUT8
DATA #CDUT9
DATA #CDUT10
DATA #DUTEOL

*

CONSPA DATA #CSPA1
DATA #CSPA2
DATA #CSPA3
DATA #CSPA4
DATA #CSPA5
DATA #CSPA6
DATA #CSPA7
DATA #CSPA8
DATA #CSPA9
DATA #CSPA10
DATA #SPAEOEOL

*

CONITA DATA #CITA1
DATA #CITA2
DATA #CITA3
DATA #CITA4
DATA #CITA5
DATA #CITA6
DATA #CITA7
DATA #CITA8
DATA #CITA9
DATA #CITA10
DATA #ITAEOL

*

CONGER DATA #CGER1
DATA #CGER2
DATA #CGER3
DATA #CGER4
DATA #CGER5
DATA #CGER6
DATA #CGER7
DATA #CGER8
DATA #CGER9
DATA #CGER10
DATA #GEREOL

*

CONFRE DATA #CFRE1
DATA #CFRE2
DATA #CFRE3

DATA #CFRE4
DATA #CFRE5
DATA #CFRE6
DATA #CFRE7
DATA #CFRE8
DATA #CFRE9
DATA #CFRE10
DATA #FREEDL

* ENGLISH CONSOLE PHRASES *

CENG1 DATA #>0224, : TEMP. :
CENG2 DATA #>0244, : WATER :
CENG3 DATA #>0264, : LEVEL :
CENG4 DATA #>0284, : CREW : :
CENG5 DATA #>0235, : SCORE : : :
CENG6 DATA #>0295, : HIGH : : :
CENG7 DATA #>00E9, : GAME :
CENG8 DATA #>0109, : OVER :
CENG9 DATA #>00E9, : PRESS :
CENG10 DATA #>0109, : REDO OR BACK :
ENGEOL

* SVENSKA CONSOLE PHRASES *

CSWD1 DATA #>0224, : TEMP. :
CSWD2 DATA #>0244, : VATTEN :
CSWD3 DATA #>0264, : NIV, : >5E, : :
CSWD4 DATA #>0284, : M, : >5B, : N : : :
CSWD5 DATA #>0235, : PO, : >5B, : NG : : :
CSWD6 DATA #>0295, : H, : >5C, : GST : : :
CSWD7 DATA #>00E9, : SPELET :
CSWD8 DATA #>0109, : SLUT :
CSWD9 DATA #>00E9, : TRYCK :
CSWD10 DATA #>0109, : REDO EL. BACK :
SWDEOL

* DUTCH CONSOLE PHRASES *

CDUT1 DATA #>0224, : TEMP. :
CDUT2 DATA #>0244, : WATER :
CDUT3 DATA #>0264, : NIVO :
CDUT4 DATA #>0284, : LEDEN :
CDUT5 DATA #>0235, : SCORE :
CDUT6 DATA #>0295, : HOOGSTE :
CDUT7 DATA #>00E9, : SPEL :
CDUT8 DATA #>0109, : VOORBIJ :
CDUT9 DATA #>00E9, : DRUK :
CDUT10 DATA #>0109, : REDO OF BACK :
DUTEOL

* SPANISH CONSOLE PHRASES *

CSPA1 DATA #>0224, : TEMP. :
CSPA2 DATA #>0244, : AGUA :
CSPA3 DATA #>0264, : NIVEL :
CSPA4 DATA #>0284, : TRIP : : :
CSPA5 DATA #>0235, : TANTEO :
CSPA6 DATA #>0295, : ALTO :
CSPA7 DATA #>00E9, : JUEGO :
CSPA8 DATA #>0109, : TERMINO :
CSPA9 DATA #>00E9, : PULSAR :

CSPA10 DATA #>0109.:REHACER O TRAS:

SPAEOI

* ITALIAN CONSOLE PHRASES *

CITA1 DATA #>0224.:TEMP.:
CITA2 DATA #>0244.:ACQUA:
CITA3 DATA #>0264.:LIV :
CITA4 DATA #>0284.:EQUIP:
CITA5 DATA #>0235.:PUNTI:::
CITA6 DATA #>0295.:ALTO :::
CITA7 DATA #>00E9.: GIOCO :
CITA8 DATA #>0109.: FINITO :
CITA9 DATA #>00E9.: PREMI :
CITA10 DATA #>0109.: REDO O BACK :
ITAEOI

* GERMAN CONSOLE PHRASES *

CGER1 DATA #>0224.:TEMP.:
CGER2 DATA #>0244.:WASSER:
CGER3 DATA #>0264.:GRAD :
CGER4 DATA #>0284.:CREW:::
CGER5 DATA #>0235.:PUNKTE:
CGER6 DATA #>0295.:HOCH :
CGER7 DATA #>00E9.: SPIEL :
CGER8 DATA #>0109.: ENDE :
CGER9 DATA #>00E9.: DR:,>5D,:CKE :
CGER10 DATA #>0109.:REDO ODER BACK:
GEREOI

* FRENCH CONSOLE PHRASES *

CFRE1 DATA #>0224.:TEMP.:
CFRE2 DATA #>0244.:EAU :
CFRE3 DATA #>0264.:NIV. :
CFRE4 DATA #>0284.:EQUIP:
CFRE5 DATA #>0235.:SCORE:::
CFRE6 DATA #>0295.:MAXI :::
CFRE7 DATA #>00E9.: PARTIE :
CFRE8 DATA #>0109.: TERMINEE :
CFRE9 DATA #>00E9.: APPUYEZ SUR :
CFRE10 DATA #>0109.: REDO OU BACK :
FREEOI

*** THE FOLLOWING MESSAGES ARE ROM ACCESSED ***

ENQPHR EQU \$

*----- WINNING MESSAGES

ADVANC DATA : ADVANCE TO NEXT LEVEL :
BONUSP DATA : BONUS POINTS GAINED! :
EXTRAS DATA : EXTRA CREW MEMBER! :

*----- LOSING MESSAGES

LASERS DATA : LASERS OVERHEATED :
MONSTE DATA : MONSTER DAMAGED SHIP :
MONSTC DATA : MONSTER ATTACKED CREW! :
SHIPLO DATA : CREW MEMBER LOST :
DETECT DATA : DETECTING UNKNOWN OBJECT :
ZYGONA DATA : ZYGONAUT APPROACHING :
CONTIN DATA : CONTINUE GAME, CAPTAIN :

*----- WATER MESSAGES

COOLAN DATA : COOLANT IS RUNNING LOW :

WATERA DATA : WATER AHEAD :
WATERT DATA : WATER TANKS FULL! :
OUTOFW DATA : OUT OF WATER, CAPTAIN! :
CONGRA DATA : CONGRATULATIONS CAPTAIN! :
PAGE

* S W E D I S H P H R A S E S *

SWDPHR EQU \$

*----- WINNING MESSAGES

SW1 DATA : FORTS:,>5B,:TT TILL N:,>5B,:STA NIV:,>5E,: :
SW2 DATA : BONUSPO:,>5B,:NG VUNNA :
SW3 DATA : EXTRA BES:,>5B,:TTNINGSMAN :

*----- LOSING MESSAGES

SW4 DATA : LASRARNA :,>5C,:VERHETTADE :
SW5 DATA : MONSTER F:,>5C,:RST:,>5C,:RDE SKIPPET:
SW6 DATA : MONSTER ANGREP BES:,>5B,:TTNING:
SW7 DATA : BES:,>5B,:TTNINGSMAN F:,>5C,:RLORAD :
SW8 DATA : OK:,>5B,:NT F:,>5C,:REM:,>5E,:L F:,>5C,:R:
DATA >5C,:VER :
SW9 DATA : SE UPP F:,>5C,:R ZYGONAUTEN :
SW10 DATA : FORTS:,>5B,:TT SPELET, KAPTEN :

*----- WATER MESSAGES

SW11 DATA : KYLV:,>5B,:TSKAN B:,>5C,:RJAR TA SLUT :
SW12 DATA : VATTEN UPPT:,>5B,:CKT :
SW13 DATA : VATTENTANKARNA FULLA :
SW14 DATA : VATTNET SLUT, KAPTEN :
SW15 DATA : GRATULERAR, KAPTEN :
PAGE

* D U T C H P H R A S E S *

DUTPHR EQU \$

*----- WINNING MESSAGES

DUT1 DATA : GA DOOR NAAR NIEUW NIVEAU :
DUT2 DATA : BONUS PUNTEN GEKREGEN :
DUT3 DATA : EXTRA BEMANNINGSLID :

*----- LOSING MESSAGES

DUT4 DATA : LASERS ZIJN OVERHIT :
DUT5 DATA : MONSTER VERNIETIGT SCHIP :
DUT6 DATA : MONSTER VIEL BEMANNING AAN:
DUT7 DATA : BEMANNINGSLID VERMIST :
DUT8 DATA : ONBEKEND VOORWERP VOOR U :
DUT9 DATA : PAS OP DE ZYGONAUT :
DUT10 DATA : GA DOOR, KAPITEIN :

*----- WATER MESSAGES

DUT11 DATA : TE WEINIG KOELVLOEISTOF :
DUT12 DATA : WATER OPGEMERKT :
DUT13 DATA : WATERTANKS VOL :
DUT14 DATA : ZONDER WATER, KAPITEIN :
DUT15 DATA : GEFELICITEERD, KAPITEIN :
PAGE

* S P A N I S H P H R A S E S *

SPAPHR EQU \$

*----- WINNING MESSAGES

SPA1 DATA : PASE AL NIVEL NUEVO :
SPA2 DATA : GANO EL BONO DE PUNTOS :
SPA3 DATA : MIEMBRO EXTRA TRIPULACION:

*----- LOSING MESSAGES

SPA4 DATA : LASERS RECALENTADOS :

SPA5 DATA : MONSTRO HA DESTRUIDO BUQUE:
SPA6 DATA : MONSTRUD HA ATACADO TRIPUL:
SPA7 DATA : PERDIDO HUMBRE TRIPULACION:
SPA8 DATA : OBJETO DESCONOCIDO DELANTE:
SPA9 DATA : JE ACERA EL "ZYGONAUT" :
SPA10 DATA : CONTINUE EL JUEGO CAPITAN:

*----- WATER MESSAGES
SPA11 DATA : BAJO NIVEL DE AGUA :
SPA12 DATA : AGUA DETECTADA :
SPA13 DATA : DEPOSITOS DE AGUA LLENOS :
SPA14 DATA : NO HAY AGUA CAPITAN :
SPA15 DATA : FELICITACIONES, CAPITAN :
PAGE

* I T A L I A N P H R A S E S *

ITAPHR EQU \$

*----- WINNING MESSAGES
ITA1 DATA : AVANZA A NUOVO LIVELLO :
ITA2 DATA : GUADAGNATI PUNTI BONUS :
ITA3 DATA : MEMBRO EQUIP. ADDIZIONALE:

*----- LOSING MESSAGES
ITA4 DATA : LASER SURRISCALDATO :
ITA5 DATA : MOSTRO HA DISTRUTTO NAVE :
ITA6 DATA : EQUIP. ATTACATO DA MOSTRO:
ITA7 DATA : MEMBRO EQUIP. PERDUTO :
ITA8 DATA : OGGETTO SCONOSCIUTO :
ITA9 DATA : ATTENTO ALLO ZYGONAUTA :
ITA10 DATA : CAPITANO, CONTINUA GIOCO :

*----- WATER MESSAGES
ITA11 DATA : RAFFREDDAMENTO RIDOTTO :
ITA12 DATA : IDENTIFICATA ACQUA :
ITA13 DATA : SERBATOI ACQUA PIENI :
ITA14 DATA : CAPITANO MANCA L'ACQUA :
ITA15 DATA : CAPITANO, CONGRATULAZIONI:
PAGE

* G E R M A N P H R A S E S *

GERPHR EQU \$

*----- WINNING MESSAGES
GER1 DATA : GEHE AUF NEUER STUFE :
GER2 DATA : ZUSATZPUNKTE GEWONNEN :
GER3 DATA : EXTRA MANNSCHAFT-MITGLIED :

*----- LOSING MESSAGES
GER4 DATA : LASER : ,>5D, : BERHITZT :
GER5 DATA : MONSTER ZERST : ,>5C, : RTE SCHIFF :
GER6 DATA : MONSTER GRIFF CREW AN :
GER7 DATA : CREW-MITGLIED VERLOREN :
GER8 DATA : UNBEKANNTES OBJEKT VORAU :
GER9 DATA : AUFPASSEN, DER ZYGONAT :
GER10 DATA : SPIEL WIETER KAPIT : ,>5B, : N :

*----- WATER MESSAGES
GER11 DATA : K : ,>5D, : HMITTEL WIRD SCHWACH :
GER12 DATA : WASSER ENTDECKT :
GER13 DATA : WASSERTANKE VOLL :
GER14 DATA : KEIN WASSER KAPIT : ,>5B, : N :
GER15 DATA : GL : ,>5D, : CKWUNSCH KAPIT : ,>5B, : N :
PAGE

* F R E N C H P H R A S E S *

FREPFR EQU \$

*----- WINNING MESSAGES

FRE1 DATA : CHANGEMENT DE NIVEAU :
FRE2 DATA : VOUS GAGNEZ LE BONUS! :
FRE3 DATA : 1 MEMBRE D'EQUIP. EN BONUS:

*----- LOSING MESSAGES

FRE4 DATA : LASERS EN SURCHAUFFE :
FRE5 DATA : VAISSEAU DETRUIT :
FRE6 DATA : EQUIPAGE ASSAILLI! :
FRE7 DATA : 1 MEMBRE D'EQUIP. PERDU :
FRE8 DATA : OBJET NON IDENTIFIE EN VUE:
FRE9 DATA : ATTENTION AU ZYGONAUTE! :
FRE10 DATA : CONTINUEZ, COMMANDANT! :

*----- WATER MESSAGES

FRE11 DATA : ATTENTION AU NIVEAU D'EAU! :
FRE12 DATA : DE L'EAU A ETE DETECTEE :
FRE13 DATA : RESERVOIRS D'EAU PLEINS! :
FRE14 DATA : PLUS D'EAU, COMMANDANT! :
FRE15 DATA : BRAVO, COMMANDANT! :

PAGE

* SIT CHARS FOR SHOWING # OF CREW MEMBERS LEFT *

TKLFT1	DATA #>20Z0, #>20Z0, #>20Z0, #>20Z0, #>20Z0	0 CREWS LEFT	
	DATA #>1B1A, #>20Z0, #>20Z0, #>20Z0, #>20Z0	1 TANK LEFT	T
	DATA #>1B1A, #>1B1A, #>20Z0, #>20Z0, #>20Z0	2 CREWS LEFT	O
	DATA #>1B1A, #>1B1A, #>1B1A, #>20Z0, #>20Z0	3 CREWS LEFT	P
	DATA #>1B1A, #>1B1A, #>1B1A, #>1B1A, #>20Z0	4 CREWS LEFT	
	DATA #>1B1A, #>1B1A, #>1B1A, #>1B1A, #>1B1A	5 CREWS LEFT	
TKLFT2	DATA #>20Z0, #>20Z0, #>20Z0, #>20Z0, #>20Z0	0 CREWS LEFT	
	DATA #>191B, #>20Z0, #>20Z0, #>20Z0, #>20Z0	1 TANK LEFT	B
	DATA #>191B, #>191B, #>20Z0, #>20Z0, #>20Z0	2 CREWS LEFT	O
	DATA #>191B, #>191B, #>191B, #>20Z0, #>20Z0	3 CREWS LEFT	T
	DATA #>191B, #>191B, #>191B, #>191B, #>20Z0	4 CREWS LEFT	
	DATA #>191B, #>191B, #>191B, #>191B, #>191B	5 CREWS LEFT	

BLANKM DATA 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2

DATA 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2

TO BLANK OUT MSGS

PAGE

* START OF GPL CODE *

STARTG MOVE 2 FROM ROM(#VDPOFF) TO @R0
MOVE 8 FROM ROM(@R0) TO VDP(0) TURN SCREEN OFF
DST >0900, @FAC LOAD IN LARGE CHARACTER SET
CALL >16 GET THE CHARACTERS
ST @WHICHL, @>7E SAVE WHICH SPEECH SET TO POINT TO

*

ST >20, @R0	START OF CPU
CLRCPU CLR *R0	CLEAR 1 BYTE OF CPURAM PAD
INC @R0	POINT TO NEXT BYTE
CEG >6A, @R0	INTO RESERVED SPOT?
BR CLRCPU	NO, SO CONTINUE
ST @>7E, @WHICHL	RESTORE SPEECH POINTER

* TITLE SCREEN GOES UP NOW *

TITLE BACK >1 BLACK BACKGROUND
ALL >20 CLEAR THE SCREEN
MOVE 2 FROM ROM(#TITCHR) TO @R0
MOVE 1152 FROM ROM(@R0) TO RAM(>0B00) GET TITLE SCREEN CHARS
MOVE 2 FROM ROM(#TITCOL) TO @R0
MOVE 32 FROM ROM(@R0) TO RAM(>0380) LOAD THE CT FOR TITLE

```

MOVE 2 FROM ROM(#TITSIT) TO @R0
MOVE 384 FROM ROM(@R0) TO RAM(>0080) PUT OU THE MM LOGO
FMT YPT=18,XPT=5, ':PRESS ANY KEY TO BEGIN:'
MOVE 2 FROM ROM(#CPYRIT) TO @R0
MOVE 8 FROM ROM(@R0) TO RAM(>908)      COPYRIGHT SYMBOL
FMT YPT=21,XPT=4, '>21', ':1983 TEXAS INSTRUMENTS:'
MOVE 2 FROM ROM(#VDPNDR) TO @R0
MOVE 8 FROM ROM(@R0) TO VDP(0)      TURN ON THE SCREEN
CLR @CHEATF      SET INITIALLY TO NO CHEAT MODE
ST 1,@DEMOFL    SET INITIALLY TO DEMO MODE
CLR @KEYBRD     SCAN ENTIRE KEYBOARD
CLR @R2        SECONDS COUNTER
S1OUTL CLR @TIMER CLEAR TIMER BYTE
SCANK1 CLR @CLRSCN SCAN THE KEYBOARD
SCAN
BS GOT1KY      WE HAVE A KEY, FOLKS!
CH 60,@TIMER  TIMER PAST 60?
BR SCANK1     NO, SO KEEP SCANNING
INC @R2      INCREMENT SECONDS
CH 5,@R2     PAST 5 SECONDS?
BR S1OUTL    NO, SO START A NEW SECOND
B @GAME     NO KEY, SO DEMO MODE FER' SURE
GOT1KY CEG >2A,@KEY '* ' KEY?
BR NODEMO   NO, SO GO AHEAD WITH THE GAME
SCANK2 CLR @CLRSCN SCAN THE KEYBOARD
SCAN
BR SCANK2   NO KEYS PRESSED YET
CEG >4A,@KEY 'J' KEY?
BS @TME     YES, SO MY CLAIM TO FAME
* CEG >50,@KEY 'P' KEY?
* BS @TPAR
CEG >23,@KEY '* ' KEY?
BR NODEMO   NO, SO GO AHEAD WITH THE GAME
SCANK3 CLR @CLRSCN SCAN THE KEYBOARD
SCAN
BR SCANK3   NO KEYS PRESSED YET
CEG >2A,@KEY '* '?
BS @TSEQ    YES, SO I GOT THE SEQUENCE
BR NODEMO   WRONG KEY HIT, SO RUN THE GAME
*
GOTSEQ ALL >20 CLEAR THE SCREEN
FMT YPT=7,XPT=11, ': TEST MODE:'
FMT YPT=11,XPT=3, ': LEVEL: ,>28,>30,>30,>2D,>35,>39,>29'
CKLVL1 CLR @CLRSCN GET A KEY
SCAN
BR CKLVL1   NOT PRESSED, SO KEEP CHECKING
$IF @KEY .LT. :0: GOTO CKLVL1
$IF @KEY .GT. :5: GOTO CKLVL1
SUB >30,@KEY SUBTRACT ASCII BIAS
ST @KEY,@R0 AND STORE IN R0
*
CKLVL2 CLR @CLRSCN GET A KEY
SCAN
BR CKLVL2   NOT PRESSED, SO KEEP CHECKING
$IF @KEY .LT. :0: GOTO CKLVL2
$IF @KEY .GT. :9: GOTO CKLVL2
M 10,@R0    MULTIPLY BY 10
SUB >30,@KEY SUBTRACT ASCII BIAS
A @KEY,@ROLB ADD IN FINAL VALUE
ST @ROLB,@GAMELV PUT INTO GAME LEVEL
*
FMT YPT=13,XPT=3, ': MOON MINERS: ,>28,>31,>2D,>39,>29'

```



```

CHKMIN CLR @CLRSCN
SCAN                                GET A KEY
BR CHKMIN                          NOT PRESSED, SO KEEP CHECKING
$IF @KEY .LT. :1: GOTO CHKMIN
$IF @KEY .GT. :9: GOTO CHKMIN
SUB >30, @KEY                      SUBTRACT ASCII BIAS
ST @KEY, @TANKS                    PUT INTO GAME TANKS
ST 1, @CHEATF                     WE HAVE CHEAT MODE!
B NODEMO                          GO PLAY THE GAME NOW!

```

```

*
GOTME ALL >20
FMT YPT=10, XPT=1, ':PROGRAMMED BY JOHN M. PHILLIPS: '
CALL WAIT20
CALL WAIT20
CALL WAIT20
CALL WAIT20
B TITLE

```

```

GOTPAR ALL >20
FMT YPT=10, XPT=2, ': TO MY PARENTS . . . . : '
FMT YPT=12, XPT=2, ':KENNETH & RUTH ANNE PHILLIPS: '
CALL WAIT20
CALL WAIT20
CALL WAIT20
CALL WAIT20
B TITLE

```

```

*****
*TEXT *          MAIN TEXT FORMATTER SUBROUTINE *
*****
*

```

```

TEXT CLR @FMT          RESET FMT
TEXT10 FETCH @FMT1    GET FORMAT CODE
$IF @FMT1 .EQ. THRU GOTO TEXTR
$IF @FMT1 .L. >EO THEN
  DSLL @FMT, 1        WORD # TIMES 2
  DST @LANG, @CNTFMT  BEGINNING OF WORD ADDRESS LIST
  DADD @FMT, @CNTFMT  ADD OFFSET INTO THE LIST
  MOVE 4 FROM ROM(@CNTFMT) TO @FMT *WORD ADDRESSES
  DSUB @FMT, @LENGTH  CALCULATE WORD LENGTH
  MOVE 2 FROM ROM(@FMT) TO @CB/ADD
  $IF @CB/ADD .DEQ. 0 GOTO TEXT
  DDECT @LENGTH
  DINCT @FMT
  MOVE @LENGTH FROM ROM(@FMT) TO RAM(@CB/ADD) *GET WORD
$END IF-THEN
$GOTO TEXT

```

```

TEXTR RTN *THRU --RTN

```

```

*****

```

```

*****
* DELAY SUBROUTINE FOR 1/2 OF A SECOND *
*****

```

```

WAIT20 CLR @TIMER    CLEAR TIMER
$REPEAT
$UNTIL @TIMER .GT. >2D
RTN

```

```

*****
* DELAY SUBROUTINE FOR 1/6 OF A SECOND *
*****

```

```

WAIT6 CLR @TIMER    CLEAR TIMER
$REPEAT
$UNTIL @TIMER .GT. >0B
RTN

```

```

*****

```

* START THE VDP INITS NOW *

```
NODEMO ST 0,@DEMOFL          WE'RE NOT IN THE DEMO MODE
GOGAME ALL >20              CLEAR THE SCREEN
    BACK >1                  BLCK BACKGROUND
    MOVE 2 FROM ROM(#VDPOFF) TO @R0
    MOVE 8 FROM ROM(@R0) TO VDP(0)
    ST >A2,@KEEPR1          TO SAVE VDP R1
    DST >900,@FAC          LOWER THIRD CHARACTER SET
    CALL >18                GET THE SMALL CHARACTER SET
```

*THIS CODE WILL FORMAT THE UPPER 2/3 OF THE SCREEN PROPERLY *

```
TOP2/3 FMT XPT=2,YPT=0,'>00',26'>02', '>03'          LINE 0
    FMT XPT=2,YPT=1,'1', '>C0',24'>C1', '>C2', '1'          LINE 1
    FMT XPT=2,YPT=2,'1', '>C7', '>C8',22'>C9', '>CA',          LINE 2
        '>C3', '1'
    FMT XPT=2,YPT=3,'1', '>C7', '>CF', '>D0',20'>D1';          LINE 3
        '>D2', '>CB', '>C3', '1'
    FMT XPT=2,YPT=4,'1', '>C7', '>CF', '>D7', '>DB',18'>D9';    LINE 4
        '>DA', '>D3', '>CB', '>C3', '1'
    FMT XPT=2,YPT=5,'1', '>C7', '>CF', '>D7', '>DF', '>E0';    LINE 5
        16'>E1', '>E2', '>DB', '>D3', '>CB', '>C3', '1'
    FMT XPT=2,YPT=6,'1', '>C7', '>CF', '>D7', '>DF', '>E7';    LINE 6
        '>E8',14'>E9', '>EA', '>E3', '>DB', '>D3';
        '>CB', '>C3', '1'
    FMT XPT=2,YPT=7,'1', '>C7', '>CF', '>D7', '>DF', '>E7';    LINE 7
        '>EF',14'>04', '>EB', '>E3', '>DB', '>D3', '>CB';
        '>C3', '1'
    FMT XPT=2,YPT=8,'1', '>C7', '>CF', '>D7', '>DF', '>E7';    LINE 8
        '>EF',14'>04', '>EB', '>E3', '>DB', '>D3', '>CB';
        '>C3', '1'
    FMT XPT=2,YPT=9,'1', '>C7', '>CF', '>D7', '>DF', '>E7';    LINE 9
        '>EE',14'>ED', '>EC', '>E3', '>DB', '>D3';
        '>CB', '>C3', '1'
    FMT XPT=2,YPT=10,'1', '>C7', '>CF', '>D7', '>DF', '>E6';    LINE 10
        16'>E5', '>E4', '>DB', '>D3', '>CB', '>C3', '1'
    FMT XPT=2,YPT=11,'1', '>C7', '>CF', '>D7', '>DE',18'>DD';    LINE 11
        '>DC', '>D3', '>CB', '>C3', '1'
    FMT XPT=2,YPT=12,'1', '>C7', '>CF', '>D6',20'>D5';          LINE 12
        '>D4', '>CB', '>C3', '1'
    FMT XPT=2,YPT=13,'1', '>C7', '>CE',22'>CD', '>CC';          LINE 13
        '>C3', '1'
    FMT XPT=2,YPT=14,'1', '>C6', '>F0',2'>F1', '>F2';          LINE 14
        '>F3',14'>C5', '>F4', '>F5',2'>F1', '>F6', '>C4';
        '1'
    FMT XPT=2,YPT=15,'>06',26'>02', '>07'          LINE 15
```

*THIS CODE WILL FORMAT THE LOWER 1/3 OF THE SCREEN PROPERLY *

```
BOT1/3 FMT XPT=2,YPT=16,'0',XPT=29,YPT=16,'3'          CORNERS
    FMT XPT=2,YPT=23,'6',XPT=29,YPT=23,'7'          CORNERS
    FMT XPT=3,YPT=16,26'2',XPT=3,YPT=23,26'2'        HORZ
    FMT XPT=2,YPT=17,6"1",XPT=29,YPT=17,6"1"        VERT
    FMT XPT=10,YPT=17,10'14',XPT=10,YPT=18,10'8'     WATER AND HEAT
    FMT XPT=9,YPT=17,2">1C",XPT=20,YPT=17,2">1D"    BORDER FOR GAGES
```

* THE FOLLOWING LINES FILL THE BOTTOM 1/3 CT,PDT,AND SIT *

```
LDSIT3 MOVE 2 FROM ROM(#COLORS) TO @R0
    MOVE 32 FROM ROM(@R0) TO RAM(>0380) PLAYING SCREEN COLORS
LDPDT9 MOVE 2 FROM ROM(#ULCORN) TO @R0
```

```

MOVE 144 FROM ROM(@R0) TO RAM(>0B00) TO PDT
MOVE 2 FROM ROM(#CREWP) TO @R0
MOVE 48 FROM ROM(@R0) TO RAM(>0B00)    CREW PATTERN
MOVE 2 FROM ROM(#AURL) TO @R0
MOVE 32 FROM ROM(@R0) TO RAM(>0ADB)    FOREIGN CHARS
MOVE 2 FROM ROM(#TCAGE1) TO @R0
MOVE 24 FROM ROM(@R0) TO RAM(>0B00)    TEMP GAGES
MOVE 2 FROM ROM(#BLANK) TO @R0
MOVE 8 FROM ROM(@R0) TO RAM(>0B40)    SIDE FILLER CHAR
*****
* CREATE MY GUNS AND BULLETS AND PLACE THEM INTO VDP RAM *
*****
CRGUNS MOVE 2 FROM ROM(#GUNS) TO @R0
MOVE 192 FROM ROM(@R0) TO RAM(>0400) TO SDT
MOVE 2 FROM ROM(#GBINIT) TO @R0
MOVE 33 FROM ROM(@R0) TO RAM(>0300) TO SAL
*
*****
* LOAD THE SOUND LISTS INTO VDP RAM *
*****
LDSND MOVE 2 FROM ROM(#FIRSND) TO @R0
MOVE 355 FROM ROM(@R0) TO RAM(>1000) TO FREE VDP4
*****
*CALL THE TEXT FORMATTER NOW TO PUT UP THE NEEDED MESSAGES ON CONSOLE *
*****
CONMSG CALL TEXT
DATA 0, 1, 2, 3, 4, 5
DATA THRU
*****
*          G O T O   R O M   C O D E          *
*****
GOROM XML >70          GOTO ROM CODE
PAGE
*****
*   G A M E   O V E R   .   .   .   R E T U R N   F R O M   R O M   C O D E   *
*****
ALDONE CEQ 1, @DEMOFL          WERE WE IN DEMO MODE?
BS STARTQ          YES, SO GO BACK TO TITLE SCREEN
CALL TEXT
DATA 6, 7          GAME OVER
DATA THRU
CALL TOPSC          CHECK FOR HIGH SCORE, IF ANY
CALL WAIT20          GAME OVER MESSAGE FLASH
CALL WAIT20          GAME OVER MESSAGE FLASH
CALL WAIT20          GAME OVER MESSAGE FLASH
CALL WAIT20          GAME OVER MESSAGE FLASH
CALL TEXT
DATA 8, 9          REDO OR BACK
DATA THRU
CLR @KEYBRD          SCAN KEYBOARD ZERO
RBLOOP CLR @CLRSCN
SCAN
CEQ >06, @KEY          REDO PRESSED?
BS GOREDO
CEQ >38, @KEY          B KEY
BS GOREDO          BACK PRESSED?
CEQ >0F, @KEY
BS GOBACK          ? KEY
CEQ >39, @KEY          ? KEY
BS GOBACK
CEQ >05, @KEY          QUIT PRESSED?
BR RBLOOP          WRONG KEY DETECTED AT THIS POINT

```

BYEBYE MOVE 2 FROM ROM(#VDPNOR) TO @R0
MOVE 8 FROM ROM(@R0) TO VDP(0)
EXIT

* COMPARE YOUR SCORE WITH BEST SCORE *

TOPSC DST >02B5, @R0 BESTSC
DST >0255, @R2 YOURSC
CEQ :*, RAM(>23B) PRESSED REDO?
BS NOTOP YES, SO NO HIGH SCORE HERE

CHKTOP CH RAM(@R2), RAM(@R0)
BS NOTOP
CH RAM(@R0), RAM(@R2)
BR NODIFF
MOVE 6 FROM RAM(>255) TO RAM(>3D00)
ST 3, @R5

SNDHI MOVE 2 FROM ROM(#BLANKS) TO @R0
MOVE 6 FROM ROM(@R0) TO RAM(>2B5) BLANK HI
CALL WAIT20
MOVE 6 FROM RAM(>3D00) TO RAM(>2B5) FLASH HI
MOVE 2 FROM ROM(#GAGSND) TO @R6
I/O @R6, 0 GONG SOUND
CALL WAIT6
DEC @R5 DONE WITH LOOP?
BR SNDHI NO, SO KEEP GOING
MOVE 6 FROM RAM(>0255) TO RAM(>02B5) NOW YOUR SCORE = HIGH SCORE
B NOTOP

NODIFF DINC @R0
DINC @R2
CEQ >02C0, @R0
BR CHKTOP

NOTOP RTN AREA

* ENTRY POINTS FOR REDO, BACK, PRESS ANY KEY STATEMENTS *

GOBACK FMT XPT=9, YPT=7, 14 '>04', XPT=9, YPT=8, 14 '>04' FILL CENTER
XML >71

B ALDONE IF PLAYER LOSES AGAIN

GOREDO FMT XPT=9, YPT=7, 14 '>04', XPT=9, YPT=8, 14 '>04' FILL CENTER
XML >72

B ALDONE IF PLAYER LOSES AGAIN

END

PAGE
GROM 4
ORC 0

* M O N M I N E *

* THIS GROM CONTAINS MONSTER AND OBJECT PATTERNS *

PRMTB1 DATA #VDPNOR, #VDPOFF, #COLORS, #TITCOL, #CPYRIT, #BLANKS
DATA #GBINIT, #ULCORN, #BLANK, #CREWP, #LGAGE, #AUML
DATA #TGAGE1, #GUNS, #FIRSND, #GAGSND

*

ZYG001 DATA >00, >00, >00, >00, >00, >00, >00, >01 THIS SPRITE
DATA >00, >00, >00, >00, >00, >00, >00, >00 IS THE LEGENDAY
DATA >00, >00, >00, >00, >00, >00, >00, >00 ZYGONAUGHT

ZYG002 DATA >00, >00, >00, >00, >00, >00, >00, >01 THIS SPRITE (IF USED)
DATA >01, >00, >00, >00, >00, >00, >00, >00 BLOB SIZE 2
DATA >00, >00, >00, >00, >00, >00, >00, >80
DATA >80, >00, >00, >00, >00, >00, >00, >00

ZYG003 DATA >00, >00, >00, >00, >00, >01, >03, >06 THIS SPRITE
DATA >07, >02, >01, >00, >00, >00, >00, >00 BLOB SIZE 1
DATA >00, >00, >00, >00, >00, >80, >C0, >60
DATA >E0, >40, >80, >00, >00, >00, >00, >00

ZYG004 DATA >00, >00, >00, >00, >01, >03, >05, >0F THIS SPRITE (IF USED)
DATA >0F, >06, >03, >01, >00, >00, >00, >00 BLOB SIZE 2
DATA >00, >00, >00, >00, >80, >C0, >A0, >F0
DATA >F0, >60, >C0, >80, >00, >00, >00, >00

ZYG005 DATA >00, >00, >00, >01, >03, >05, >0E, >1F THIS SPRITE (IF USED)
DATA >1F, >0C, >07, >03, >01, >00, >00, >00 BLOB SIZE 3
DATA >00, >00, >00, >80, >C0, >A0, >70, >F1
DATA >F1, >3B, >E0, >C0, >80, >00, >00, >00

ZYG006 DATA >00, >00, >03, >07, >0B, >1D, >3D, >3F THIS SPRITE
DATA >3F, >3D, >1B, >0D, >07, >03, >00, >00 BLOB SIZE 4
DATA >00, >00, >C0, >E0, >D0, >BB, >BC, >FC
DATA >FC, >5C, >0B, >50, >E0, >C0, >00, >00

ZYG007 DATA >00, >03, >07, >0F, >1B, >3D, >7D, >7F THIS SPRITE (IF USED)
DATA >7F, >7A, >30, >1A, >0F, >07, >03, >00 BLOB SIZE 5
DATA >00, >C0, >E0, >F0, >DB, >BC, >BE, >FE
DATA >FE, >AE, >04, >AB, >F0, >E0, >C0, >00

ZYG008 DATA >03, >0F, >1F, >37, >73, >F1, >F9, >FF THIS SPRITE
DATA >FF, >F5, >E0, >75, >3F, >1F, >0F, >03 ZYGO FULL W/
DATA >C0, >F0, >FB, >EC, >CE, >8F, >9F, >FF MOUTH CLOSED
DATA >FF, >57, >03, >56, >FC, >FB, >F0, >C0

ZYG009 DATA >03, >0F, >1F, >37, >73, >F1, >F9, >FF THIS SPRITE
DATA >FB, >F0, >E0, >70, >3B, >1F, >0F, >03 ZYGO FULL W/
DATA >C0, >F0, >FB, >EC, >CE, >8F, >9F, >FF MOUTH OPENED
DATA >0F, >07, >03, >06, >0C, >FB, >F0, >C0

ZYG010 DATA >03, >0F, >1F, >3F, >7F, >FF, >FF, >FF THIS SPRITE
DATA >FF, >FF, >FF, >7F, >3F, >1F, >0F, >03 ZYGO SIDE VIEW
DATA >C0, >F0, >FB, >EC, >F6, >FE, >FE, >FE
DATA >FC, >E0, >C0, >E2, >FC, >FB, >F0, >C0

***** THAT WAS 320 BYTES OF THE 'LEGENDARY ZYGONAUGHT' SPRITE

ZYG011 DATA >00, >00, >00, >00, >00, >00, >00, >01 THIS SPRITE
DATA >00, >00, >00, >00, >00, >00, >00, >00 IS THE LEGENDAY
DATA >00, >00, >00, >00, >00, >00, >00, >00 ZYGONAUGHT EYES
DATA >00, >00, >00, >00, >00, >00, >00, >00

ZYG012 DATA >00, >00, >00, >00, >00, >00, >00, >01 THIS SPRITE (IF USED)

```

DATA >01,>00,>00,>00,>00,>00,>00,>00      EYES SIZE 2
DATA >00,>00,>00,>00,>00,>00,>00,>80
DATA >80,>00,>00,>00,>00,>00,>00,>00
ZYGD13 DATA >00,>00,>00,>00,>00,>00,>00,>01      THIS SPRITE
DATA >00,>01,>00,>00,>00,>00,>00,>00      EYES SIZE 3
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>80,>00,>00,>00,>00,>00
ZYGD14 DATA >00,>00,>00,>00,>00,>00,>02,>00      THIS SPRITE (IF USED)
DATA >00,>00,>00,>00,>00,>00,>00,>00      EYES SIZE 4
DATA >00,>00,>00,>00,>00,>00,>40,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
ZYGD15 DATA >00,>00,>00,>00,>00,>02,>01,>00      THIS SPRITE (IF USED)
DATA >00,>00,>00,>00,>00,>00,>00,>00      EYES SIZE 5
DATA >00,>00,>00,>00,>00,>40,>80,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
ZYGD16 DATA >00,>00,>00,>00,>04,>02,>02,>00      THIS SPRITE
DATA >00,>00,>00,>00,>00,>00,>00,>00      EYES SIZE 6
DATA >00,>00,>00,>00,>20,>40,>40,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
ZYGD17 DATA >00,>00,>00,>00,>04,>02,>02,>00      THIS SPRITE (IF USED)
DATA >00,>00,>00,>00,>00,>00,>00,>00      EYES SIZE 7
DATA >00,>00,>00,>00,>20,>40,>40,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
ZYGD18 DATA >00,>00,>00,>08,>0C,>0E,>06,>00      THIS SPRITE
DATA >00,>00,>00,>00,>00,>00,>00,>00      ZYGO FULL W/
DATA >00,>00,>00,>10,>30,>70,>60,>00      MOUTH CLOSED
DATA >00,>00,>00,>00,>00,>00,>00,>00
ZYGD19 DATA >00,>00,>00,>08,>04,>02,>06,>00      THIS SPRITE
DATA >00,>00,>00,>00,>00,>00,>00,>00      ZYGO FULL W/
DATA >00,>00,>00,>10,>30,>70,>60,>00      MOUTH OPENED
DATA >00,>00,>00,>00,>00,>00,>00,>00
ZYGD20 DATA >00,>00,>00,>00,>00,>00,>00,>00      THIS SPRITE
DATA >00,>00,>00,>00,>00,>00,>00,>00      ZYGO SIDE VIEW
DATA >00,>00,>00,>10,>08,>00,>00,>00      OF EYES
DATA >00,>00,>00,>00,>00,>00,>00,>00
***** THAT WAS 320 BYTES OF THE 'ZYGONAUGHT EYES' SPRITE
*****
* CREW MEMBER ANIMATION DEFINITIONS(FROM FIRING SQUAD) *
*****
CREWUP DATA >01,>01,>01,>03,>07,>0F,>0B,>0B      THIS SPRITE
DATA >0B,>0B,>07,>06,>06,>06,>06,>0C      CREW FIRING
DATA >80,>80,>20,>90,>D0,>F0,>B0,>80      AWAY FROM ME
DATA >80,>80,>C0,>C0,>C0,>C0,>C0,>60
CREWUL DATA >01,>01,>01,>03,>07,>0F,>0B,>0B      THIS SPRITE
DATA >0B,>0B,>07,>0E,>00,>00,>00,>00      CREW FIRING
DATA >80,>80,>20,>90,>D0,>F0,>B0,>80      W/ LEFT LEG UP
DATA >80,>80,>C0,>C0,>C0,>C0,>C0,>60
CREWUR DATA >01,>01,>01,>03,>07,>0F,>0B,>0B      THIS SPRITE
DATA >0B,>0B,>07,>06,>06,>06,>06,>0C      CREW FIRING
DATA >80,>80,>20,>90,>D0,>F0,>B0,>80      W/ RIGHT LEG UP
DATA >80,>80,>C0,>E0,>00,>00,>00,>00
CREWDN DATA >01,>01,>20,>11,>17,>1F,>1B,>03      THIS SPRITE
DATA >03,>0B,>07,>06,>06,>06,>06,>0C      CREW FIRING
DATA >80,>80,>80,>80,>C0,>E0,>A0,>A0      TOWARDS ME
DATA >A0,>A0,>C0,>C0,>C0,>C0,>C0,>60
CREWDL DATA >01,>01,>20,>11,>17,>1F,>1B,>03      THIS SPRITE
DATA >03,>03,>07,>0E,>00,>00,>00,>00      CREW FIRING
DATA >80,>80,>80,>80,>C0,>E0,>A0,>A0      TOWARDS ME

```

```

DATA >A0,>A0,>C0,>C0,>C0,>C0,>C0,>60
CREWDR DATA >01,>01,>20,>11,>17,>1F,>1B,>03
DATA >03,>0B,>07,>06,>06,>06,>06,>0C
DATA >80,>80,>80,>80,>C0,>E0,>A0,>A0
DATA >A0,>A0,>C0,>E0,>00,>00,>00,>00
CREWCR DATA >00,>00,>00,>07,>0B,>0F,>07,>07
DATA >07,>03,>03,>03,>03,>03,>03,>03
DATA >C0,>C0,>80,>C0,>C0,>80,>40,>00
DATA >00,>00,>00,>00,>00,>00,>00,>80
CREWMR DATA >00,>00,>00,>07,>0B,>17,>17,>0F
DATA >0F,>0B,>0B,>04,>09,>12,>22,>33
DATA >C0,>C0,>80,>C0,>C0,>D0,>60,>00
DATA >00,>80,>80,>80,>00,>00,>00,>00
CREWOR DATA >00,>00,>00,>0F,>13,>27,>47,>0F
DATA >0F,>0D,>0C,>FB,>80,>00,>00,>00
DATA >C0,>C0,>80,>C0,>C0,>AB,>10,>00
DATA >00,>80,>C0,>40,>40,>20,>20,>30

```

```

W/ LEFT LEG UP
THIS SPRITE
CREW FIRING
TOWARDS ME
W/ RIGHT LEG UP
THIS SPRITE
CREW CLOSED RIGHT
THIS SPRITE
CREW MEDIUM RIGHT
THIS SPRITE
CREW OPEN RIGHT

```

****===> THAT FINISHES THE CREW MEMBER RIGHT ANIMATION PATTERNS

```

CREWCL DATA >03,>03,>01,>03,>03,>01,>02,>00
DATA >00,>00,>00,>00,>00,>00,>00,>01
DATA >00,>00,>00,>E0,>D0,>F0,>E0,>E0
DATA >E0,>C0,>C0,>C0,>C0,>C0,>C0,>C0
CREWML DATA >03,>03,>01,>03,>03,>0B,>06,>00
DATA >00,>01,>01,>01,>00,>00,>00,>00
DATA >00,>00,>00,>E0,>D0,>EB,>EB,>F0
DATA >F0,>B0,>B0,>20,>90,>4B,>44,>CC
CREWOL DATA >03,>03,>01,>03,>23,>15,>0B,>00
DATA >00,>01,>03,>02,>02,>04,>04,>0C
DATA >00,>00,>00,>F0,>CB,>E4,>E2,>F0
DATA >F0,>B0,>30,>1F,>01,>00,>00,>00

```

```

THIS SPRITE
CREW CLOSED LEFT
THIS SPRITE
CREW MEDIUM LEFT
THIS SPRITE
CREW OPEN LEFT

```

****===> THAT FINISHES THE CREW MEMBER LEFT ANIMATION PATTERNS

* OBJECT DEFINITIONS (2X,UNMAG) [BIX SIZES PER OBJECT] *

```

WATER1 DATA >00,>00,>07,>0F,>07,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>E0,>F0,>E0,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
WATER2 DATA >00,>00,>1F,>3F,>1F,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>F0,>FB,>F0,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
WATER3 DATA >00,>00,>1F,>7F,>3F,>1F,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>E0,>F0,>FB,>F0,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
WATER4 DATA >00,>19,>3F,>7F,>3F,>1F,>02,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>80,>F0,>FB,>F0,>FB,>30,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
WATER5 DATA >0B,>1D,>3F,>7F,>7F,>3F,>1B,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>90,>FB,>FC,>FB,>FC,>7B,>10
DATA >00,>00,>00,>00,>00,>00,>00,>00
WATER6 DATA >18,>3D,>7F,>7F,>7F,>7F,>3B,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >10,>BB,>FC,>FE,>FC,>FE,>7C,>4B
DATA >00,>00,>00,>00,>00,>00,>00,>00

```

WATER7 DATA >18,>7D,>7F,>7F,>FF,>7F,>3B,>31
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >30,>B8,>FE,>FF,>FE,>FF,>7E,>6C
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 WATER8 DATA >38,>7D,>FF,>7F,>FF,>FF,>7B,>31
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >30,>BC,>FE,>FF,>FE,>FF,>7F,>6C
 DATA >00,>00,>00,>00,>00,>00,>00,>00

*****=> 256 BYTES OF THE WATER OBJECT

TREAS EQU \$ TREASURES
 SILVR1 DATA >00,>00,>00,>04,>0E,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>00,>40,>E0,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 SILVR2 DATA >00,>00,>00,>0C,>1E,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>00,>60,>F0,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 SILVR3 DATA >00,>00,>00,>1C,>3E,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>00,>70,>FB,>F0,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 SILVR4 DATA >00,>00,>08,>1C,>3E,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>20,>70,>FB,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 SILVR5 DATA >00,>00,>08,>1C,>3E,>7F,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>20,>70,>FB,>FC,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 SILVR6 DATA >00,>18,>3C,>7E,>FF,>7E,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>18,>3C,>7E,>FF,>7E,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 SILVR7 DATA >08,>1C,>3E,>7F,>FF,>7F,>3E,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >10,>38,>7C,>FE,>FF,>FE,>7C,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 SILVR8 DATA >08,>1C,>3E,>7F,>FF,>7F,>3E,>1C
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >10,>38,>7C,>FE,>FF,>FE,>7C,>38
 DATA >00,>00,>00,>00,>00,>00,>00,>00

*****=> 256 BYTES OF THE SILVER OBJECT

JADES1 DATA >00,>00,>00,>18,>18,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>00,>18,>18,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 JADES2 DATA >00,>00,>10,>18,>18,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>08,>18,>18,>10,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 JADES3 DATA >00,>00,>18,>18,>18,>18,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>18,>18,>18,>18,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 JADES4 DATA >00,>00,>1C,>1C,>1C,>1C,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>38,>38,>38,>38,>00,>00

DATA >00,>00,>00,>00,>00,>00,>00,>00
 JADES5 DATA >00,>08,>1C,>3E,>3E,>1C,>08,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>10,>3B,>7C,>7C,>3B,>10,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 JADES6 DATA >00,>08,>1C,>3E,>7F,>3E,>1C,>08
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>10,>3B,>7C,>FE,>7C,>3B,>10
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 JADES7 DATA >00,>08,>1C,>3E,>7F,>3E,>1C,>08
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>10,>3B,>7C,>FE,>7C,>3B,>10
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 JADES8 DATA >00,>08,>1C,>36,>62,>36,>1C,>08
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>10,>3B,>6C,>46,>6C,>3B,>10
 DATA >00,>00,>00,>00,>00,>00,>00,>00

*****=> 256 BYTES OF THE JADE OBJECT

AMETH1 DATA >00,>00,>00,>05,>0F,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>00,>40,>E0,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 AMETH2 DATA >00,>00,>05,>0D,>1F,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>40,>60,>F0,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 AMETH3 DATA >00,>00,>09,>1F,>3F,>1B,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>20,>F0,>FB,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 AMETH4 DATA >00,>11,>3B,>7F,>00,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>10,>BB,>FC,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 AMETH5 DATA >00,>10,>11,>3B,>7F,>21,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>10,>10,>BB,>FC,>08,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 AMETH6 DATA >00,>10,>31,>3B,>7B,>33,>00,>02
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>10,>1B,>BB,>BC,>9B,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 AMETH7 DATA >00,>08,>31,>7B,>FB,>33,>11,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>30,>1B,>BC,>BE,>9B,>10,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 AMETH8 DATA >00,>1B,>39,>7B,>FB,>3B,>11,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>30,>3B,>BC,>BE,>BB,>10,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00

*****=> 256 BYTES OF THE AMETHYST OBJECT

* MONSTER DEFINITIONS (2X,UNMAG) [8 SIZES PER MONSTER]

MONSO1 DATA >00,>00,>00,>00,>00,>00,>00,>01
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00
 DATA >00,>00,>00,>00,>00,>00,>00,>00

THIS SPRITE
BFLY SIZE 1

```

MONS02 DATA >00,>00,>00,>00,>00,>00,>00,>01          THIS SPRITE (IF USED)
        DATA >01,>00,>00,>00,>00,>00,>00,>00          BFLY SIZE 2
        DATA >00,>00,>00,>00,>00,>00,>00,>80
        DATA >80,>00,>00,>00,>00,>00,>00,>00
MONS03 DATA >00,>00,>00,>00,>00,>00,>01,>03          THIS SPRITE
        DATA >03,>01,>00,>00,>00,>00,>00,>00          BFLY SIZE 1
        DATA >00,>00,>00,>00,>00,>00,>80,>C0
        DATA >C0,>80,>00,>00,>00,>00,>00,>00
MONS04 DATA >00,>00,>00,>00,>00,>01,>05,>07          THIS SPRITE (IF USED)
        DATA >07,>05,>01,>00,>00,>00,>00,>00          BFLY SIZE 2
        DATA >00,>00,>00,>00,>00,>80,>A0,>E0
        DATA >E0,>A0,>80,>00,>00,>00,>00,>00
MONS05 DATA >00,>00,>00,>02,>03,>09,>0D,>0F          THIS SPRITE (IF USED)
        DATA >0F,>0F,>09,>03,>01,>00,>00,>00          BFLY SIZE 3
        DATA >00,>00,>00,>40,>C0,>90,>80,>F0
        DATA >F0,>F0,>90,>C0,>80,>00,>00,>00
MONS06 DATA >00,>00,>04,>02,>03,>39,>3D,>0F          THIS SPRITE
        DATA >0F,>3F,>39,>03,>01,>00,>00,>00          BFLY SIZE 4
        DATA >00,>00,>20,>40,>C0,>9C,>BC,>F0
        DATA >F0,>FC,>9C,>C0,>A0,>00,>00,>00
MONS07 DATA >08,>08,>05,>03,>63,>39,>3D,>0F          THIS SPRITE (IF USED)
        DATA >0F,>3F,>39,>63,>05,>09,>08,>00          BFLY SIZE 5
        DATA >10,>10,>A0,>E0,>C6,>9C,>BC,>F0
        DATA >F0,>FC,>9C,>C6,>A0,>90,>10,>00
MONS08 DATA >38,>08,>07,>06,>E3,>F9,>3D,>1F          THIS SPRITE
        DATA >1F,>3F,>F9,>E3,>05,>09,>0A,>38          BFLY SIZE 4
        DATA >1C,>D0,>E0,>60,>C7,>9F,>BC,>FB
        DATA >FB,>FC,>9F,>C7,>A0,>90,>50,>1C
***** THAT WAS 256 BYTES OF THE 'BFLY' SPRITE          !!!!!
MONS11 DATA >00,>00,>00,>00,>00,>00,>00,>01          THIS SPRITE
        DATA >00,>00,>00,>00,>00,>00,>00,>00          PEGS SIZE 1
        DATA >00,>00,>00,>00,>00,>00,>00,>00
        DATA >00,>00,>00,>00,>00,>00,>00,>00
MONS12 DATA >00,>00,>00,>00,>00,>00,>00,>01          THIS SPRITE (IF USED)
        DATA >01,>00,>00,>00,>00,>00,>00,>00          PEGS SIZE 2
        DATA >00,>00,>00,>00,>00,>00,>00,>80
        DATA >80,>00,>00,>00,>00,>00,>00,>00
MONS13 DATA >00,>00,>00,>00,>00,>00,>02,>01          THIS SPRITE
        DATA >01,>02,>00,>00,>00,>00,>00,>00          PEGS SIZE 3
        DATA >00,>00,>00,>00,>00,>00,>40,>80
        DATA >80,>40,>00,>00,>00,>00,>00,>00
MONS14 DATA >00,>00,>00,>00,>00,>06,>02,>01          THIS SPRITE (IF USED)
        DATA >01,>02,>02,>00,>00,>00,>00,>00          PEGS SIZE 4
        DATA >00,>00,>00,>00,>00,>60,>40,>80
        DATA >80,>40,>40,>00,>00,>00,>00,>00
MONS15 DATA >00,>00,>00,>00,>18,>1C,>05,>03          THIS SPRITE (IF USED)
        DATA >03,>03,>04,>04,>08,>00,>00,>00          PEGS SIZE 5
        DATA >00,>00,>00,>00,>18,>38,>A0,>C0
        DATA >C0,>C0,>20,>20,>10,>00,>00,>00
MONS16 DATA >00,>00,>10,>38,>7C,>45,>05,>07          THIS SPRITE
        DATA >03,>03,>03,>0D,>0C,>0C,>10,>00          PEGS SIZE 6
        DATA >00,>00,>08,>1C,>3E,>A2,>A0,>E0
        DATA >C0,>C0,>C0,>B0,>30,>30,>08,>00
MONS17 DATA >00,>10,>38,>7C,>FD,>CD,>8D,>0F          THIS SPRITE (IF USED)
        DATA >07,>07,>07,>0D,>0C,>0C,>30,>00          PEGS SIZE 7
        DATA >00,>08,>1C,>3E,>BF,>B3,>B1,>F0
        DATA >E0,>E0,>E0,>B0,>30,>30,>0C,>00
MONS18 DATA >18,>3C,>7E,>FC,>FD,>CD,>8D,>BF          THIS SPRITE

```

```

DATA >07,>07,>07,>0D,>0D,>0D,>78,>78          PEGS SIZE 8
DATA >18,>3C,>7E,>3F,>BF,>B3,>B1,>F1
DATA >E0,>E0,>E0,>80,>80,>80,>1E,>1E
***** THAT WAS 256 BYTES OF THE 'PEGASUS' SPRITE !!!!!
MONS21 DATA >00,>00,>00,>00,>00,>00,>00,>01    THIS SPRITE
DATA >00,>00,>00,>00,>00,>00,>00,>00          EGGS SIZE 1
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
MONS22 DATA >00,>00,>00,>00,>00,>00,>00,>01    THIS SPRITE (IF USED)
DATA >01,>00,>00,>00,>00,>00,>00,>00          EGGS SIZE 2
DATA >00,>00,>00,>00,>00,>00,>00,>80
DATA >80,>00,>00,>00,>00,>00,>00,>00
MONS23 DATA >00,>00,>00,>00,>00,>00,>04,>01    THIS SPRITE
DATA >01,>04,>00,>00,>00,>00,>00,>00          EGGS SIZE 3
DATA >00,>00,>00,>00,>00,>00,>20,>80
DATA >80,>20,>00,>00,>00,>00,>00,>00
MONS24 DATA >00,>00,>00,>00,>04,>02,>01,>03    THIS SPRITE (IF USED)
DATA >01,>02,>04,>00,>00,>00,>00,>00          EGGS SIZE 4
DATA >00,>00,>00,>00,>10,>A0,>C0,>60
DATA >C0,>A0,>10,>00,>00,>00,>00,>00
MONS25 DATA >00,>00,>00,>00,>00,>01,>1B,>0E    THIS SPRITE (IF USED)
DATA >03,>01,>00,>00,>00,>00,>00,>00          EGGS SIZE 5
DATA >00,>00,>00,>00,>00,>C0,>EC,>88
DATA >E0,>C0,>80,>80,>80,>00,>00,>00
MONS26 DATA >00,>00,>00,>00,>01,>03,>32,>2F    THIS SPRITE
DATA >02,>03,>01,>00,>00,>01,>00,>00          EGGS SIZE 6
DATA >00,>00,>00,>00,>C0,>E0,>A6,>FA
DATA >20,>E0,>C0,>80,>80,>40,>00,>00
MONS27 DATA >00,>00,>00,>00,>01,>03,>62,>5F    THIS SPRITE (IF USED)
DATA >42,>03,>03,>01,>00,>00,>07,>00          EGGS SIZE 7
DATA >00,>00,>00,>00,>C0,>E0,>A3,>FD
DATA >21,>E0,>E0,>C0,>80,>80,>70,>00
MONS28 DATA >00,>00,>00,>01,>03,>04,>76,>5F    THIS SPRITE
DATA >5C,>05,>07,>03,>00,>00,>0F,>0F          EGGS SIZE 8
DATA >00,>00,>00,>C0,>E0,>90,>B7,>FD
DATA >1D,>D0,>F0,>E0,>80,>80,>78,>78
***** THAT WAS 256 BYTES OF THE 'EGGHEAD' SPRITE !!!!!
MONS31 DATA >00,>00,>00,>00,>00,>00,>00,>01    THIS SPRITE
DATA >00,>00,>00,>00,>00,>00,>00,>00          CRAB SIZE 1
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
MONS32 DATA >00,>00,>00,>00,>00,>00,>00,>01    THIS SPRITE (IF USED)
DATA >01,>00,>00,>00,>00,>00,>00,>00          CRAB SIZE 2
DATA >00,>00,>00,>00,>00,>00,>00,>80
DATA >80,>00,>00,>00,>00,>00,>00,>00
MONS33 DATA >00,>00,>00,>00,>00,>00,>02,>01    THIS SPRITE
DATA >01,>02,>00,>00,>00,>00,>00,>00          CRAB SIZE 3
DATA >00,>00,>00,>00,>00,>00,>40,>80
DATA >80,>40,>00,>00,>00,>00,>00,>00
MONS34 DATA >00,>00,>00,>00,>04,>08,>07,>03    THIS SPRITE (IF USED)
DATA >03,>07,>08,>04,>00,>00,>00,>00          CRAB SIZE 4
DATA >00,>00,>00,>00,>20,>10,>E0,>C0
DATA >C0,>E0,>10,>20,>00,>00,>00,>00
MONS35 DATA >00,>00,>00,>08,>10,>17,>0F,>06    THIS SPRITE (IF USED)
DATA >05,>0F,>17,>10,>08,>00,>00,>00          CRAB SIZE 5
DATA >00,>00,>00,>10,>08,>EB,>F0,>A0
DATA >60,>F0,>EB,>08,>10,>00,>00,>00
MONS36 DATA >00,>00,>10,>20,>2F,>3F,>10,>08    THIS SPRITE

```

```

DATA >0A,>1F,>3F,>2F,>20,>10,>00,>00          CRAB SIZE 6
DATA >00,>00,>08,>04,>F4,>FC,>5B,>10
DATA >B0,>FB,>FC,>F4,>04,>08,>00,>00
MONS37 DATA >00,>18,>30,>60,>6F,>7F,>3D,>08          THIS SPRITE (IF USED)
DATA >0A,>3F,>7F,>6F,>60,>30,>18,>00          CRAB SIZE 7
DATA >00,>18,>0C,>06,>F6,>FE,>5C,>10
DATA >B0,>FC,>FE,>F6,>06,>0C,>18,>00
MONS38 DATA >38,>60,>C0,>DF,>DF,>FF,>7A,>10          THIS SPRITE
DATA >10,>75,>FF,>DF,>DF,>C0,>60,>3B          CRAB SIZE 8
DATA >1C,>06,>03,>FB,>FB,>FF,>AE,>08
DATA >08,>5F,>FF,>FB,>FB,>03,>06,>1C          BY CURTIS KLING!
***** THAT WAS 256 BYTES OF THE 'CRAB' SPRITE          !!!!!
MONS41 DATA >00,>00,>00,>00,>00,>00,>00,>01          THIS SPRITE
DATA >00,>00,>00,>00,>00,>00,>00,>00          ROBY SIZE 1
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
MONS42 DATA >00,>00,>00,>00,>00,>00,>00,>01          THIS SPRITE (IF USED)
DATA >01,>00,>00,>00,>00,>00,>00,>00          ROBY SIZE 2
DATA >00,>00,>00,>00,>00,>00,>00,>80
DATA >80,>00,>00,>00,>00,>00,>00,>00
MONS43 DATA >00,>00,>00,>00,>00,>00,>03,>02          THIS SPRITE
DATA >01,>01,>00,>00,>00,>00,>00,>00          ROBY SIZE 3
DATA >00,>00,>00,>00,>00,>00,>00,>40
DATA >80,>80,>00,>00,>00,>00,>00,>00
MONS44 DATA >00,>00,>00,>00,>00,>06,>05,>05          THIS SPRITE (IF USED)
DATA >03,>02,>01,>00,>00,>00,>00,>00          ROBY SIZE 4
DATA >00,>00,>00,>00,>00,>60,>A0,>A0
DATA >C0,>40,>80,>00,>00,>00,>00,>00
MONS45 DATA >00,>00,>00,>18,>0E,>0B,>09,>05          THIS SPRITE (IF USED)
DATA >07,>05,>02,>02,>01,>00,>00,>00          ROBY SIZE 5
DATA >00,>00,>00,>18,>70,>D0,>90,>A0
DATA >E0,>A0,>40,>40,>80,>00,>00,>00
MONS46 DATA >00,>00,>20,>38,>26,>23,>31,>3B          THIS SPRITE
DATA >3F,>16,>0B,>0D,>04,>07,>00,>00          ROBY SIZE 6
DATA >00,>00,>0B,>3B,>6B,>CB,>9B,>BB
DATA >F8,>D0,>A0,>60,>40,>C0,>00,>00
MONS47 DATA >00,>60,>3B,>2E,>26,>26,>72,>7B          THIS SPRITE (IF USED)
DATA >7F,>56,>1B,>09,>00,>04,>07,>00          ROBY SIZE 7
DATA >00,>0C,>3B,>EB,>CB,>CB,>9C,>BC
DATA >FC,>D4,>B0,>20,>60,>40,>C0,>00
MONS48 DATA >80,>E0,>7B,>2E,>26,>66,>72,>7B          THIS SPRITE
DATA >7F,>76,>DB,>D9,>86,>06,>06,>03          ROBY SIZE 8
DATA >01,>07,>1E,>74,>64,>66,>4E,>DE
DATA >FE,>6E,>DB,>9B,>B1,>B0,>30,>E0          BY MIKE ROBICHEAUX
***** THAT WAS 256 BYTES OF THE 'ROBYNTH' SPRITE          !!!!!
MONS51 DATA >00,>00,>00,>00,>00,>00,>00,>01          THIS SPRITE
DATA >00,>00,>00,>00,>00,>00,>00,>00          HYDRA SIZE 1
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
MONS52 DATA >00,>00,>00,>00,>00,>00,>00,>01          THIS SPRITE (IF USED)
DATA >01,>00,>00,>00,>00,>00,>00,>00          HYDRA SIZE 2
DATA >00,>00,>00,>00,>00,>00,>00,>80
DATA >80,>00,>00,>00,>00,>00,>00,>00
MONS53 DATA >00,>00,>00,>00,>00,>00,>02,>02          THIS SPRITE
DATA >01,>03,>00,>00,>00,>00,>00,>00          HYDRA SIZE 3
DATA >00,>00,>00,>00,>00,>00,>40,>40
DATA >80,>C0,>00,>00,>00,>00,>00,>00
MONS54 DATA >00,>00,>00,>00,>04,>0E,>0E,>04          THIS SPRITE (IF USED)

```



```

DATA >01,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>80
DATA >80,>00,>00,>00,>00,>00,>00,>00
MONS73 DATA >00,>00,>00,>00,>00,>00,>02,>01
DATA >03,>01,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>40,>80
DATA >C0,>00,>00,>00,>00,>00,>00,>00
MONS74 DATA >00,>00,>00,>00,>00,>00,>02,>01
DATA >03,>02,>01,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>40,>80
DATA >C0,>40,>80,>00,>00,>00,>00,>00
MONS75 DATA >00,>00,>00,>00,>00,>08,>07,>05
DATA >03,>02,>01,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>20,>40,>C0
DATA >80,>80,>00,>00,>00,>00,>00,>00
MONS76 DATA >00,>00,>00,>00,>10,>0B,>0F,>0D
DATA >07,>06,>03,>01,>00,>00,>00,>00
DATA >00,>00,>00,>00,>10,>A0,>E0,>60
DATA >C0,>C0,>80,>00,>00,>00,>00,>00
MONS77 DATA >00,>00,>08,>10,>10,>1F,>1B,>0D
DATA >0F,>0F,>05,>06,>03,>01,>00,>00
DATA >00,>00,>20,>10,>10,>F0,>A0,>60
DATA >E0,>E0,>40,>C0,>80,>00,>00,>00
MONS78 DATA >20,>40,>4F,>5F,>73,>79,>7B,>1D
DATA >1F,>0F,>0C,>06,>07,>03,>01,>00
DATA >0B,>04,>E4,>F4,>9C,>3C,>BC,>70
DATA >F0,>E0,>60,>C0,>C0,>80,>00,>00
***** THAT WAS 256 BYTES OF THE ' DEVIL ' SPRITE !!!!!
MONS81 DATA >00,>00,>00,>00,>00,>00,>00,>01
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
MONS82 DATA >00,>00,>00,>00,>00,>00,>00,>01
DATA >01,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>80
DATA >80,>00,>00,>00,>00,>00,>00,>00
MONS83 DATA >00,>00,>00,>00,>00,>00,>01,>03
DATA >01,>01,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>80,>C0
DATA >80,>80,>00,>00,>00,>00,>00,>00
MONS84 DATA >00,>00,>00,>00,>00,>01,>03,>07
DATA >01,>03,>05,>09,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>80,>C0,>E0
DATA >80,>C0,>A0,>90,>00,>00,>00,>00
MONS85 DATA >00,>00,>00,>00,>01,>03,>05,>0F
DATA >01,>03,>05,>09,>19,>00,>00,>00
DATA >00,>00,>00,>00,>80,>C0,>A0,>F0
DATA >80,>C0,>A0,>90,>9B,>00,>00,>00
MONS86 DATA >00,>00,>00,>00,>01,>03,>05,>0F
DATA >01,>03,>05,>09,>19,>3B,>00,>00
DATA >00,>00,>00,>00,>80,>C0,>A0,>F0
DATA >80,>C0,>A0,>90,>9B,>DC,>00,>00
MONS87 DATA >00,>00,>00,>00,>01,>03,>05,>0F
DATA >01,>03,>07,>0D,>19,>3B,>7B,>00
DATA >00,>00,>00,>00,>80,>C0,>A0,>F0
DATA >80,>C0,>E0,>80,>9B,>DC,>DE,>00
MONS88 DATA >00,>00,>00,>00,>01,>03,>05,>0F
DATA >01,>03,>07,>0D,>19,>31,>73,>F3

```

```

DATA >00,>00,>00,>00,>80,>C0,>A0,>F0
DATA >80,>C0,>E0,>B0,>98,>BC,>CE,>CE
***** THAT WAS 256 BYTES OF THE 'TETRA' SPRITE
MONS91 DATA >00,>00,>00,>00,>00,>00,>00,>01
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
MONS92 DATA >00,>00,>00,>00,>00,>00,>00,>01
DATA >01,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>80
DATA >80,>00,>00,>00,>00,>00,>00,>00
MONS93 DATA >00,>00,>00,>00,>00,>00,>00,>03
DATA >01,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>C0
DATA >80,>00,>00,>00,>00,>00,>00,>00
MONS94 DATA >00,>00,>00,>00,>00,>00,>01,>07
DATA >01,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>80,>E0
DATA >80,>00,>00,>00,>00,>00,>00,>00
MONS95 DATA >00,>00,>00,>00,>00,>00,>01,>0F
DATA >11,>02,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>80,>F0
DATA >88,>40,>00,>00,>00,>00,>00,>00
MONS96 DATA >00,>00,>00,>00,>02,>01,>01,>0F
DATA >1B,>23,>02,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>40,>80,>80,>F0
DATA >DB,>C4,>40,>00,>00,>00,>00,>00
MONS97 DATA >00,>00,>08,>07,>05,>03,>01,>1F
DATA >3B,>73,>62,>42,>00,>00,>00,>00
DATA >00,>00,>10,>E0,>A0,>C0,>80,>FB
DATA >DC,>CE,>46,>42,>00,>00,>00,>00
MONS98 DATA >20,>10,>0F,>0D,>07,>02,>01,>1F
DATA >3F,>7B,>F2,>E2,>C5,>80,>80,>00
DATA >04,>0B,>F0,>B0,>E0,>40,>80,>FB
DATA >FC,>DE,>4F,>47,>A3,>01,>01,>00
***** THAT WAS 256 BYTES OF THE 'BATMAN
MONSA1 DATA >00,>00,>00,>00,>00,>00,>00,>01
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>00
MONSA2 DATA >00,>00,>00,>00,>00,>00,>00,>01
DATA >01,>00,>00,>00,>00,>00,>00,>00
DATA >00,>00,>00,>00,>00,>00,>00,>80
DATA >80,>00,>00,>00,>00,>00,>00,>00
MONSA3 DATA >00,>00,>00,>03,>07,>05,>07,>07
DATA >06,>03,>08,>07,>00,>00,>00,>00
DATA >00,>00,>00,>C0,>E0,>A0,>E0,>E0
DATA >60,>C0,>10,>E0,>00,>00,>00,>00
MONSA4 DATA >00,>03,>07,>0F,>09,>0F,>0E,>0F
DATA >06,>03,>01,>10,>08,>07,>00,>00
DATA >00,>C0,>E0,>F0,>90,>F0,>F0,>F0
DATA >60,>C0,>80,>08,>10,>E0,>00,>00
MONSA5 DATA >0F,>1F,>1F,>39,>39,>3F,>1E,>0F
DATA >04,>07,>23,>31,>28,>24,>03,>00
DATA >F0,>FB,>FB,>9C,>9C,>FC,>78,>F0
DATA >20,>E0,>C4,>8C,>14,>24,>C0,>00
MONSA6 DATA >0F,>1F,>3F,>31,>39,>3F,>3E,>1F
DATA >1E,>25,>23,>33,>28,>24,>23,>00

```

BY CURTIS KLING!

!!!!

THIS SPRITE
BAT SIZE 1THIS SPRITE (IF USED)
BAT SIZE 2THIS SPRITE
BAT SIZE 3THIS SPRITE (IF USED)
BAT SIZE 4THIS SPRITE (IF USED)
BAT SIZE 5THIS SPRITE
BAT SIZE 6THIS SPRITE (IF USED)
BAT SIZE 7THIS SPRITE
BAT SIZE 8

' SPRITE !!!!!

THIS SPRITE
SKULL SIZE 1THIS SPRITE (IF USED)
SKULL SIZE 2THIS SPRITE
SKULL SIZE 3THIS SPRITE (IF USED)
SKULL SIZE 4THIS SPRITE (IF USED)
SKULL SIZE 5THIS SPRITE
SKULL SIZE 6

DATA >AB,>AB,>AB

FOR TITLE SCREEN

DATA >1A,>1A,>1A,>1A,>1A,>1A,>1A

DATA >F1,>F1,>F1

*

CPYRIT DATA >3C,>42,>99,>A1,>A1,>99,>42,>3C

COPYRIGHT SYMBOL

BLANKS DATA >20,>20,>20,>20,>20,>20

BLANKS FOR SCORE

* SAL INITIAL ATTRIBUTES FOR GUNS, BULS, AND MONS *

GBINIT DATA >FF,>78,>80,>06,>6F,>78,>84,>06

TOP N BOT

DATA >38,>18,>88,>06,>38,>DB,>8C,>06

SIDES

DATA >C0,>01,>90,>06,>C0,>FF,>94,>06

BULLETS

DATA >C0,>01,>98,>04,>C0,>01,>9C,>08,>D0

TWO MONSTERS

* DEFINE CHARS FOR SCREEN

*

ULCORN DATA >FF,>C0,>A0,>90,>88,>84,>82,>81

CHAR 0

VERTBN DATA >81,>81,>81,>81,>81,>81,>81,>81

CHAR 1

HORZBN DATA >FF,0,0,0,0,0,0,>FF

CHAR 2

URCORN DATA >FF,>03,5,9,>11,>21,>41,>81

CHAR 3

BLID DATA >FF,>FF,>FF,>FF,>FF,>FF,>FF,>FF

CHAR 4

BLANK DATA >00,>00,>00,>00,>00,>00,>00,>00

CHAR 5 AND CHAR >68

LLCORN DATA >81,>82,>84,>88,>90,>A0,>C0,>FF

CHAR 6

LRCORN DATA >81,>41,>21,>11,>09,>05,>03,>FF

CHAR 7

WGAGE1 DATA >00,>FF,>FF,>FF,>FF,>FF,>FF,>00

CHAR 8 WATER GAGE 1

WGAGE2 DATA >00,>FE,>FE,>FE,>FE,>FE,>FE,>00

CHAR 9 WATER GAGE 2

WGAGE3 DATA >00,>FC,>FC,>FC,>FC,>FC,>FC,>00

CHAR 10 WATER GAGE 3

WGAGE4 DATA >00,>FB,>FB,>FB,>FB,>FB,>FB,>00

CHAR 11 WATER GAGE 4

WGAGE5 DATA >00,>F0,>F0,>F0,>F0,>F0,>F0,>00

CHAR 12 WATER GAGE 5

WGAGE6 DATA >00,>E0,>E0,>E0,>E0,>E0,>E0,>00

CHAR 13 WATER GAGE 6

WGAGE7 DATA >00,>C0,>C0,>C0,>C0,>C0,>C0,>00

CHAR 14 WATER GAGE 7

WGAGE8 DATA >00,>80,>80,>80,>80,>80,>80,>00

CHAR 15 WATER GAGE 8

WGAGE9 DATA >00,>00,>00,>00,>00,>00,>00,>00

CHAR 16 WATER GAGE 9

CONSOL DATA >FF,>FF,>FF,>FF,>FF,>FF,>FF,>FF

CHAR 17 CONSOLE FILL CHAR

*** THAT WAS 144 BYTES

**** THESE NEXT SIX CHARACTERS GO INTO PDT @>BC0 (CHAR SET 3) *

CREWP DATA >01,>01,>01,>03,>07,>0F,>0B,>0B

THIS SPRITE

DATA >0B,>0B,>07,>06,>06,>06,>06,>0C

CREW FIRING

DATA >80,>80,>20,>90,>D0,>F0,>B0,>80

AWAY FROM ME

DATA >80,>80,>C0,>C0,>C0,>C0,>C0,>60

LGAGE DATA >01,>01,>01,>01,>01,>01,>01,>01

CHAR 28 LEFT GAGE BORDER

ROAGE DATA >80,>80,>80,>80,>80,>80,>80,>80

CHAR 29 RIGHT GAGE BORDER

**** THESE NEXT 3 CHARACTERS GO INTO PDT @>ADB (CHAR SET 11)*

AUML DATA >00,>88,>00,>20,>50,>F8,>88,>88

CHAR >5B A UMLAUT

OUML DATA >00,>88,>00,>70,>88,>88,>88,>70

CHAR >5C O UMLAUT

UUML DATA >00,>88,>00,>90,>90,>90,>90,>68

CHAR >5D U UMLAUT

AUMLS DATA >00,>20,>00,>20,>50,>F8,>88,>88

CHAR >5E A UMLAUT SWEDE

**** THESE NEXT 3 CHARACTERS GO INTO PDT @>B00 (CHAR SET 12)*

TGAGE1 DATA >00,>FF,>FF,>FF,>FF,>FF,>FF,>00

CHAR 96 TEMP. GAGE 1

TGAGE2 DATA >00,>00,>00,>00,>00,>00,>00,>00

CHAR 97 TEMP. GAGE 2

```

REDSQR DATA >FF,>FF,>FF,>FF,>FF,>FF,>FF CHAR 98 RED SQUARE
*****
* GUN, BULLET DEFINITIONS (2X, UNMAG) *
*****
GUNS DATA >00,>00,>00,>00,>00,>00,>00,>00 WILL BE CHAR >80
DATA >1F,>0B,>04,>07,>02,>03,>01,>01 THIS SPRITE
DATA >00,>00,>00,>00,>00,>00,>00,>00
DATA >FB,>D0,>20,>E0,>40,>C0,>80,>80 TOP GUN
*
DATA >01,>01,>03,>02,>07,>04,>0B,>1F THIS SPRITE (IF USED)
DATA >00,>00,>00,>00,>00,>00,>00,>00 WILL BE CHAR >84
DATA >80,>80,>C0,>40,>E0,>20,>D0,>FB BOTTOM GUN
DATA >00,>00,>00,>00,>00,>00,>00,>00
*
DATA >00,>00,>00,>80,>C0,>80,>DC,>D7 THIS SPRITE (IF USED)
DATA >D7,>DC,>80,>C0,>80,>00,>00,>00 WILL BE CHAR >88
DATA >00,>00,>00,>00,>00,>00,>00,>00 LEFT GUN
DATA >00,>00,>00,>00,>00,>00,>00,>00
*
DATA >00,>00,>00,>00,>00,>00,>00,>00 THIS SPRITE
DATA >00,>00,>00,0,0,0,0,0 WILL BE CHAR >8C
DATA >00,>00,>00,>01,>03,>0D,>3B,>EB RIGHT GUN
DATA >EB,>3B,>0D,>03,>01,>00,>00,>00
*
DATA >00,>00,>00,>00,>00,>01,>01,>01 THIS SPRITE (IF USED)
DATA >01,>01,>01,>00,>00,>00,>00,>00 WILL BE CHAR >90
DATA >00,>00,>00,>00,>00,>00,>00,>00 UP DOWN BULLET
DATA >00,>00,>00,>00,>00,>00,>00,>00
*
DATA >00,>00,>00,>00,>00,>00,>00,>07 THIS SPRITE
DATA >00,>00,>00,>00,>00,>00,>00,>00 WILL BE CHAR >94
DATA >00,>00,>00,>00,>00,>00,>00,>E0 RIGHT LEFT BULLET
DATA 0,0,0,0,0,0,0,0

```

***** THAT WAS 192 BYTES OF THE SIX STATIC SPRITES

* START OF VDP SOUND LISTS *

FIRSND DATA 4,>9A,>BF,>DF,>FF,1 >1000 - >105B 94 BYTES

DATA 3,>8F,>28,>9A,1

DATA 3,>8F,>24,>99,1

DATA 3,>8F,>20,>98,1

ORIGINALLY

DATA 3,>8F,>1C,>97,1

94 BYTES

DATA 3,>8F,>18,>96,1

DATA 3,>8F,>14,>95,1

DATA 3,>8F,>10,>94,1

DATA 3,>8F,>0C,>93,1

DATA 3,>8F,>0A,>92,1

DATA 3,>8F,>0C,>93,1

DATA 3,>8F,>10,>94,1

DATA 3,>8F,>14,>95,1

DATA 3,>8F,>18,>96,1

DATA 3,>8F,>1C,>97,1

DATA 3,>8F,>20,>98,1

DATA 3,>8F,>24,>99,1

DATA 3,>8F,>28,>9A,1

DATA 1,>9F,0

*
EXPSND DATA 7,>9F,>BF,>DF,>E7,>F0,>C0,>07,5 >105B - >1091

DATA 1, >F1, 6
DATA 1, >F2, 7
DATA 1, >F3, 8
DATA 1, >F4, 9
DATA 1, >F5, 10
DATA 1, >F6, 11
DATA 1, >F7, 12
DATA 1, >F8, 13
DATA 1, >F9, 14
DATA 1, >FA, 15
DATA 1, >FB, 16
DATA 1, >FC, 17
DATA 1, >FD, 18
DATA 1, >FE, 30
DATA 1, >FF, 0

54 BYTES

*
TNKSND DATA 4, >9F, >BA, >DF, >FF, 1
DATA 3, >A9, >3F, >B2, 4
DATA 1, >BF, 0

TANK MOVING SOUND
14 BYTES >1092 - >1099

*
WLKSND DATA 4, >9F, >BA, >DF, >FF, 1
DATA 3, >A0, >OE, >B6, 1
DATA 1, >BF, 0

CREW MOVING SOUND 14 BYTES

*
SNDOFF DATA 4, >9F, >BF, >DF, >FF, 0

TURN ALL SOUNDS OFF 6 BYTES

*
ZYCSND DATA 4, >9F, >BF, >DF, >FA, 1
DATA 2, >E4, >FE, 4
DATA 2, >E4, >FC, 4
DATA 2, >E4, >FA, 4
DATA 2, >E4, >FB, 4
DATA 2, >E4, >F6, 3
DATA 2, >E4, >F4, 3
DATA 2, >E4, >F2, 3
DATA 2, >E4, >F0, 3
DATA 1, >FF, 0

ZYGONAUGHT COMING SOUND
41 BYTES

*
ZYGSND DATA 4, >9F, >BF, >DF, >FA, 1
DATA 2, >E4, >F0, 2
DATA 2, >E4, >F2, 2
DATA 2, >E4, >F4, 2
DATA 2, >E4, >F6, 2
DATA 2, >E4, >F8, 3
DATA 2, >E4, >FA, 3
DATA 2, >E4, >FC, 3
DATA 2, >E4, >FE, 3
DATA 1, >FF, 0

ZYGONAUGHT COMING SOUND
41 BYTES

*
HITSND DATA 4, >9F, >BF, >DF, >FA, 1
DATA 2, >E5, >F0, 1
DATA 1, >FF, 0

13 BYTES

*
GAGSND DATA 4, >9A, >BF, >DF, >FF, 1
DATA 3, >86, >0D, >90, 4
DATA 3, >86, >0D, >93, 5
DATA 3, >86, >0D, >96, 6
DATA 3, >86, >0D, >99, 7
DATA 3, >86, >0D, >9C, 8

WATER GAGE FULL SOUND

39 BYTES

DATA 3, >86, >0D, >9E, 9
DATA 1, >9F, 0

*

BONSND DATA 4, >9A, >BF, >DF, >FF, 1
DATA 3, >87, >2A, >90, 4
DATA 3, >87, >2A, >93, 5
DATA 3, >87, >2A, >96, 6
DATA 3, >87, >2A, >99, 7
DATA 3, >87, >2A, >9C, 8
DATA 3, >87, >2A, >9E, 9
DATA 1, >9F, 0
END

BONUS POINTS SOUND

39 BYTES

PAGE
GROM 5*****
* THIS GROM RESERVED FOR SPEECH DATA *

```

* ADVANCE TO NEW LEVEL 1
DATA #270
DATA >41,>8F,>C6,>DC,>37,>59,>67,>BD ADVANCE T
DATA >EA,>49,>D7,>C6,>AC,>2B,>19,>47 ADVANCE T
DATA >93,>59,>5E,>AB,>7B,>95,>55,>63 ADVANCE T
DATA >1B,>A9,>16,>E1,>76,>8D,>E3,>95 ADVANCE T
DATA >4A,>5B,>39,>2D ADVANCE T
DATA >66,>0A,>6A,>9E,>E5,>7B,>E8,>39 ADVANCE T
DATA >AB,>79,>B7,>E2,>A1,>97,>68,>61 ADVANCE T
DATA >9D,>76,>86,>56,>4A,>8B,>8D,>CB ADVANCE T
DATA >29,>4A,>C9,>11,>DA,>26,>A7,>2B ADVANCE T
DATA >B9,>85,>E9,>8B ADVANCE T
DATA >92,>2C,>1B,>D9,>49,>99,>4B,>8D ADVANCE T
DATA >2A,>00,>AD,>32,>04,>60,>49,>8A ADVANCE T
DATA >00,>86,>49,>45,>80,>0B,>EE,>00 ADVANCE T
DATA >06,>18,>36,>DD,>01,>C7,>9A,>06 ADVANCE T
DATA >B5,>67,>E7,>80 ADVANCE T
DATA >77,>1A,>B4,>6B,>3C,>2A,>DE,>5D ADVANCE T
DATA >D2,>A2,>0F,>77,>3F,>A7,>41,>8B ADVANCE T
DATA >39,>5C,>E3,>9C,>34,>2D,>6A,>8B ADVANCE T
DATA >8B,>55,>B2,>B4,>12,>DD,>CC,>5B ADVANCE T
DATA >F1,>D6,>96,>5A ADVANCE T
DATA >B3,>AB,>CB,>DB,>D9,>6A,>53,>6A ADVANCE T
DATA >13,>0D,>7F,>27,>46,>C9,>DE,>34 ADVANCE T
DATA >3D,>95,>32,>25,>F7,>72,>D7,>70 ADVANCE T
DATA >8B,>80,>EE,>46,>39,>80,>01,>04 ADVANCE T
DATA >74,>51,>A6,>80 ADVANCE T
DATA >63,>4D,>3D,>C0,>34,>63,>3B,>DB ADVANCE T
DATA >26,>71,>50,>B5,>1F,>A7,>7C,>C7 ADVANCE T
DATA >59,>2B,>9A,>85,>95,>54,>A7,>74 ADVANCE T
DATA >E8,>DE,>77,>4B,>AF,>DC,>A9,>0F ADVANCE T
DATA >3D,>84,>0A,>A9 ADVANCE T
DATA >95,>33,>F4,>10,>2A,>25,>57,>49 ADVANCE T
DATA >31,>7C,>AC,>14,>5A,>D9,>4B,>57 ADVANCE T
DATA >2C,>BC,>34,>D2,>32,>DD,>94,>09 ADVANCE T
DATA >B2,>56,>E5,>1D,>E7,>AB,>35,>22 ADVANCE T
DATA >31,>4A,>94,>2B ADVANCE T
DATA >4F,>B3,>66,>3A,>56,>AE,>3C,>C5 ADVANCE T
DATA >92,>E1,>9B,>D9,>32,>0F,>AB,>A4 ADVANCE T
DATA >E5,>07 ADVANCE T
* BONUS POINTS GAINED! 2
DATA #222
DATA >49,>55,>A9,>82,>EB,>1B,>17,>D5 BONUS POI
DATA >A5,>0A,>AE,>63,>DC,>54,>1F,>2A BONUS POI
DATA >94,>CF,>49,>57,>8B,>D2,>C5,>C6 BONUS POI
DATA >67,>53,>A1,>DE,>76,>12,>97,>04 BONUS POI
DATA >B7,>67,>24,>2E BONUS POI
DATA >5C,>32,>61,>A6,>95,>3B,>F1,>C5 BONUS POI
DATA >84,>6B,>B6,>62,>C7,>67,>17,>4E BONUS POI
DATA >EB,>4C,>BC,>6B,>80,>E2,>CA,>14 BONUS POI
DATA >50,>74,>2B,>03,>0A,>76,>03,>02 BONUS POI
DATA >04,>13,>66,>00 BONUS POI
DATA >E7,>DD,>0B,>80,>86,>8B,>D1,>F5 BONUS POI

```

DATA	>2C, >13, >6B, >3B, >C5, >F0, >A9, >C2	BONUS POI
DATA	>BB, >A2, >36, >23, >45, >77, >97, >8E	BONUS POI
DATA	>D2, >8C, >12, >D5, >2C, >5B, >4E, >33	BONUS POI
DATA	>A2, >17, >F3, >7A	BONUS POI
DATA	>C6, >46, >57, >D6, >D6, >DB, >69, >33	BONUS POI
DATA	>5D, >D3, >F0, >F2, >B0, >4B, >75, >05	BONUS POI
DATA	>1C, >E3, >CE, >80, >EC, >4D, >39, >80	BONUS POI
DATA	>80, >29, >4A, >31, >10, >94, >D5, >4A	BONUS POI
DATA	>93, >93, >49, >51	BONUS POI
DATA	>5A, >14, >F5, >7E, >C7, >4D, >CE, >56	BONUS POI
DATA	>AA, >36, >DD, >4C, >A9, >49, >2D, >89	BONUS POI
DATA	>47, >87, >92, >2A, >55, >A5, >AA, >59	BONUS POI
DATA	>CF, >A6, >B3, >B2, >96, >EC, >74, >26	BONUS POI
DATA	>C2, >8A, >1A, >3E	BONUS POI
DATA	>CD, >58, >33, >DB, >48, >F9, >15, >75	BONUS POI
DATA	>42, >1D, >64, >33, >9E, >55, >91, >89	BONUS POI
DATA	>90, >63, >A5, >AB, >18, >39, >25, >56	BONUS POI
DATA	>E4, >A8, >36, >A2, >24, >98, >31, >3A	BONUS POI
DATA	>AA, >8D, >63, >64	BONUS POI
DATA	>2A, >1F, >EE, >6A, >B4, >1F	BONUS POI
* EXTRA	CREW MEMBER!	3
DATA	#200	
DATA	>49, >2B, >C9, >5D, >62, >D3, >34, >2D	EXTRA CRE
DATA	>67, >33, >ED, >76, >DA, >D5, >AA, >36	EXTRA CRE
DATA	>B5, >A6, >54, >F7, >50, >6A, >80, >D6	EXTRA CRE
DATA	>4A, >05, >F0, >73, >3A, >02, >7E, >4F	EXTRA CRE
DATA	>E3, >40, >00, >9C	EXTRA CRE
DATA	>F1, >0A, >42, >EE, >D3, >15, >6E, >3B	EXTRA CRE
DATA	>09, >39, >86, >29, >9B, >DB, >22, >85	EXTRA CRE
DATA	>6C, >1E, >1A, >B5, >AB, >18, >25, >30	EXTRA CRE
DATA	>40, >18, >E5, >0E, >28, >A1, >C3, >01	EXTRA CRE
DATA	>25, >4F, >25, >A6	EXTRA CRE
DATA	>A6, >2E, >63, >92, >32, >DB, >98, >53	EXTRA CRE
DATA	>39, >28, >F1, >E0, >42, >0A, >37, >8D	EXTRA CRE
DATA	>CA, >93, >EB, >7C, >10, >82, >96, >2C	EXTRA CRE
DATA	>D5, >68, >41, >B2, >46, >3B, >D9, >E3	EXTRA CRE
DATA	>04, >D9, >18, >DD	EXTRA CRE
DATA	>E2, >90, >3A, >94, >E0, >CD, >33, >CA	EXTRA CRE
DATA	>EE, >D2, >52, >34, >F3, >48, >25, >4B	EXTRA CRE
DATA	>4F, >D1, >CD, >B3, >E4, >0C, >23, >24	EXTRA CRE
DATA	>0B, >F3, >B4, >EB, >4C, >E5, >74, >5C	EXTRA CRE
DATA	>2B, >89, >33, >A5	EXTRA CRE
DATA	>B7, >2D, >29, >C7, >CA, >14, >41, >3F	EXTRA CRE
DATA	>A5, >1C, >39, >CB, >58, >D3, >B4, >48	EXTRA CRE
DATA	>12, >6C, >5B, >25, >43, >A6, >4A, >B2	EXTRA CRE
DATA	>6D, >E1, >49, >DD, >AA, >C1, >B1, >95	EXTRA CRE
DATA	>BB, >64, >AB, >05	EXTRA CRE
DATA	>D7, >75, >AA, >90, >8E, >EB, >5C, >DB	EXTRA CRE
DATA	>B9, >52, >2B, >8A, >71, >5D, >41, >0D	EXTRA CRE
DATA	>8E, >28, >D2, >7E	EXTRA CRE
* GOOD	SHOT, CAPTAIN	4
DATA	#217	
DATA	>4E, >1D, >B1, >C3, >2C, >65, >05, >35	GOOD SHOT
DATA	>5A, >8B, >90, >8B, >53, >95, >21, >65	GOOD SHOT
DATA	>1B, >16, >32, >B1, >97, >98, >5D, >98	GOOD SHOT
DATA	>D9, >C6, >19, >42, >B1, >EE, >E6, >EB	GOOD SHOT
DATA	>C4, >48, >5A, >84	GOOD SHOT
DATA	>67, >36, >58, >A3, >44, >AC, >E1, >99	GOOD SHOT

DATA	>76,>14,>D0,>21,>19,>01,>FE,>1E	GOOD SHOT
DATA	>73,>C0,>5F,>17,>08,>F8,>F3,>54	GOOD SHOT
DATA	>01,>7F,>AE,>1A,>E0,>CF,>B5,>21	GOOD SHOT
DATA	>55,>1D,>6E,>3A	GOOD SHOT
DATA	>71,>97,>98,>52,>99,>DA,>D9,>59	GOOD SHOT
DATA	>7C,>48,>55,>9E,>A5,>64,>70,>8E	GOOD SHOT
DATA	>54,>46,>8C,>92,>21,>84,>96,>E9	GOOD SHOT
DATA	>D1,>6E,>9B,>E2,>5B,>86,>67,>2A	GOOD SHOT
DATA	>49,>4A,>2B,>61	GOOD SHOT
DATA	>BA,>CD,>86,>00,>41,>84,>12,>60	GOOD SHOT
DATA	>2A,>55,>03,>2C,>11,>86,>80,>20	GOOD SHOT
DATA	>42,>13,>90,>B5,>8D,>03,>BA,>76	GOOD SHOT
DATA	>4A,>FA,>48,>A1,>65,>25,>A6,>99	GOOD SHOT
DATA	>A9,>78,>46,>84	GOOD SHOT
DATA	>E2,>66,>C7,>1A,>E9,>D1,>8A,>83	GOOD SHOT
DATA	>25,>82,>8B,>8B,>AB,>06,>40,>40	GOOD SHOT
DATA	>74,>AE,>60,>80,>41,>DD,>93,>EE	GOOD SHOT
DATA	>B4,>A7,>F7,>2A,>2E,>7A,>50,>9E	GOOD SHOT
DATA	>96,>AF,>2A,>E9	GOOD SHOT
DATA	>DE,>8B,>7B,>AC,>2C,>A7,>19,>ED	GOOD SHOT
DATA	>1D,>51,>8C,>8C,>66,>84,>8D,>89	GOOD SHOT
DATA	>8B,>56,>9A,>E6,>F6,>96,>E2,>5A	GOOD SHOT
DATA	>AB,>42,>FB,>7B,>9A,>13,>A5,>0B	GOOD SHOT
DATA	>1D,>E7,>A6,>4E	GOOD SHOT
DATA	>1E,>00	GOOD SHOT

* MONSTER DESTROYED

5

DATA	#210	
DATA	>44,>D5,>5C,>BB,>B5,>9C,>1A,>D3	MONSTER D
DATA	>48,>1B,>E3,>F2,>D1,>74,>5F,>87	MONSTER D
DATA	>43,>53,>F1,>D4,>A7,>3A,>54,>17	MONSTER D
DATA	>D3,>D9,>D3,>49,>50,>8C,>8C,>17	MONSTER D
DATA	>93,>44,>46,>35	MONSTER D
DATA	>D6,>D3,>30,>BC,>0B,>20,>99,>34	MONSTER D
DATA	>01,>14,>53,>2E,>80,>E9,>CB,>0B	MONSTER D
DATA	>30,>B9,>08,>28,>E0,>85,>B0,>22	MONSTER D
DATA	>15,>EF,>26,>51,>55,>8B,>D4,>25	MONSTER D
DATA	>A3,>64,>4D,>29	MONSTER D
DATA	>9D,>91,>A4,>96,>D4,>E2,>52,>8C	MONSTER D
DATA	>5C,>49,>97,>8B,>54,>8D,>A6,>F8	MONSTER D
DATA	>D9,>51,>C0,>76,>E1,>1A,>50,>C0	MONSTER D
DATA	>F0,>E1,>0A,>DB,>3E,>4D,>00,>2F	MONSTER D
DATA	>96,>0B,>E0,>D7	MONSTER D
DATA	>34,>04,>FC,>E6,>A6,>80,>BA,>3C	MONSTER D
DATA	>0D,>30,>FA,>64,>31,>5D,>77,>D7	MONSTER D
DATA	>E4,>7A,>DD,>98,>FA,>D4,>97,>EE	MONSTER D
DATA	>4A,>99,>CB,>45,>D9,>FA,>34,>8A	MONSTER D
DATA	>39,>AC,>98,>52	MONSTER D
DATA	>43,>C7,>49,>73,>52,>72,>09,>5F	MONSTER D
DATA	>A5,>C5,>89,>DE,>3C,>B2,>99,>24	MONSTER D
DATA	>37,>78,>8D,>DB,>50,>1A,>5C,>A7	MONSTER D
DATA	>AD,>AB,>96,>89,>72,>8D,>52,>4B	MONSTER D
DATA	>5B,>A9,>CB,>B5	MONSTER D
DATA	>2B,>22,>C3,>1D,>1B,>E3,>84,>14	MONSTER D
DATA	>D7,>B0,>CD,>8C,>9E,>4A,>94,>DE	MONSTER D
DATA	>09,>D3,>47,>EE,>30,>9C,>38,>44	MONSTER D
DATA	>1F,>A5,>22,>A0,>9D,>3C	MONSTER D

* WAY TO GO, CAPTAIN

6

DATA #222

DATA	>4A,>E2,>27,>C4,>BC,>ED,>16,>4D	WAY TO GO
DATA	>E7,>52,>95,>AD,>BA,>B4,>10,>DC	WAY TO GO
DATA	>4C,>D7,>EE,>51,>4B,>34,>33,>5B	WAY TO GO
DATA	>A7,>43,>49,>56,>23,>E6,>E4,>14	WAY TO GO
DATA	>B5,>6B,>27,>97	WAY TO GO
DATA	>B2,>B3,>E4,>1A,>4C,>22,>DF,>F1	WAY TO GO
DATA	>90,>5A,>16,>AE,>3B,>47,>4A,>F5	WAY TO GO
DATA	>1A,>F2,>2D,>E9,>20,>95,>F3,>AE	WAY TO GO
DATA	>CA,>72,>1C,>00,>67,>BC,>96,>DB	WAY TO GO
DATA	>B3,>73,>DB,>3B	WAY TO GO
DATA	>5D,>62,>34,>1E,>15,>EF,>EE,>8B	WAY TO GO
DATA	>D1,>87,>BB,>9F,>D3,>23,>C5,>1C	WAY TO GO
DATA	>AE,>71,>4E,>96,>1C,>52,>B8,>D9	WAY TO GO
DATA	>39,>59,>4A,>0C,>11,>2A,>6D,>77	WAY TO GO
DATA	>68,>31,>A4,>1B	WAY TO GO
DATA	>6D,>95,>66,>7B,>1F,>E1,>72,>51	WAY TO GO
DATA	>93,>E9,>82,>59,>DB,>4B,>75,>A6	WAY TO GO
DATA	>B3,>AA,>23,>6F,>1B,>20,>01,>59	WAY TO GO
DATA	>DB,>3B,>A0,>6B,>A7,>A4,>8F,>14	WAY TO GO
DATA	>5A,>56,>62,>9A	WAY TO GO
DATA	>99,>8A,>67,>44,>2B,>6E,>76,>AC	WAY TO GO
DATA	>91,>1E,>AD,>3B,>5B,>22,>8B,>8B	WAY TO GO
DATA	>8B,>6A,>00,>04,>44,>E7,>0A,>06	WAY TO GO
DATA	>1B,>D4,>3D,>E9,>4E,>7B,>7A,>AF	WAY TO GO
DATA	>E2,>A2,>07,>E5	WAY TO GO
DATA	>49,>F9,>AA,>92,>EE,>8D,>BB,>C7	WAY TO GO
DATA	>CA,>72,>9A,>D1,>DE,>11,>C5,>CB	WAY TO GO
DATA	>68,>46,>DB,>9B,>8B,>6B,>A5,>69	WAY TO GO
DATA	>6E,>6F,>29,>AE,>85,>2A,>84,>BF	WAY TO GO
DATA	>A7,>39,>11,>0A	WAY TO GO
DATA	>B5,>19,>9D,>2D,>FB,>01	WAY TO GO

* LASERS OVERHEATED

7

DATA	#290	
DATA	>C2,>E4,>2E,>C3,>2F,>69,>07,>53	LASERS OV
DATA	>FA,>8C,>F8,>52,>12,>74,>63,>C3	LASERS OV
DATA	>FD,>C7,>49,>D3,>4A,>36,>91,>4B	LASERS OV
DATA	>27,>5B,>3B,>6A,>F1,>6A,>BA,>AB	LASERS OV
DATA	>65,>A9,>D9,>4B	LASERS OV
DATA	>E4,>BD,>9A,>A5,>14,>A3,>56,>F5	LASERS OV
DATA	>8A,>9A,>5C,>92,>AA,>EE,>D1,>0A	LASERS OV
DATA	>B2,>57,>16,>19,>83,>DB,>A8,>5F	LASERS OV
DATA	>A8,>BB,>B4,>3D,>A3,>7C,>5F,>62	LASERS OV
DATA	>EC,>52,>9B,>D0	LASERS OV
DATA	>9D,>AA,>A9,>D7,>6D,>42,>CE,>2E	LASERS OV
DATA	>A1,>19,>B7,>09,>B9,>3A,>A7,>64	LASERS OV
DATA	>DD,>22,>E4,>66,>12,>1A,>71,>8A	LASERS OV
DATA	>5B,>AA,>2A,>67,>C4,>2E,>52,>B6	LASERS OV
DATA	>1A,>1A,>15,>25	LASERS OV
DATA	>CA,>46,>FE,>B9,>CD,>D5,>68,>2B	LASERS OV
DATA	>F9,>07,>77,>11,>91,>8D,>14,>11	LASERS OV
DATA	>DC,>45,>44,>B6,>D2,>82,>2F,>0D	LASERS OV
DATA	>FB,>36,>95,>5F,>5A,>0C,>A5,>CE	LASERS OV
DATA	>6B,>A7,>A9,>B6	LASERS OV
DATA	>BA,>9B,>8F,>D2,>A4,>9A,>EA,>66	LASERS OV
DATA	>32,>76,>8D,>EE,>94,>9B,>8B,>B9	LASERS OV
DATA	>51,>5A,>62,>2E,>61,>EE,>2C,>6B	LASERS OV
DATA	>AE,>BB,>14,>9B,>92,>A2,>FA,>A1	LASERS OV
DATA	>B2,>94,>4D,>8A	LASERS OV

DATA	>62,>B3,>C4,>6B,>D7,>49,>8A,>0D	LASERS OV
DATA	>12,>EF,>49,>27,>A9,>CD,>2A,>A6	LASERS OV
DATA	>95,>94,>A0,>9F,>DE,>AA,>10,>96	LASERS OV
DATA	>9D,>36,>2C,>61,>E4,>38,>6E,>4A	LASERS OV
DATA	>77,>2C,>D9,>4D	LASERS OV
DATA	>AB,>A9,>33,>29,>5A,>8E,>E2,>A4	LASERS OV
DATA	>CE,>2B,>68,>B5,>8A,>85,>6E,>98	LASERS OV
DATA	>C6,>64,>29,>42,>40,>D3,>E9,>16	LASERS OV
DATA	>10,>C0,>4E,>4E,>C5,>CD,>C6,>2C	LASERS OV
DATA	>FC,>15,>3B,>57	LASERS OV
DATA	>A6,>E8,>9E,>62,>6C,>BD,>E4,>79	LASERS OV
DATA	>ED,>91,>79,>8A,>4D,>5C,>54,>22	LASERS OV
DATA	>93,>D6,>12,>49,>F2,>4E,>CA,>1D	LASERS OV
DATA	>DA,>18,>D1,>A9,>69,>9D,>2A,>61	LASERS OV
DATA	>16,>25,>AA,>F2	LASERS OV
DATA	>8A,>1E	LASERS OV

* MONSTER DAMAGED SHIP

DATA	#233	B
DATA	>4C,>91,>D2,>B7,>24,>DD,>24,>C3	MONSTER D
DATA	>4B,>59,>63,>F7,>95,>74,>D3,>5A	MONSTER D
DATA	>4D,>47,>71,>D5,>86,>EA,>72,>9A	MONSTER D
DATA	>5B,>39,>E9,>4A,>51,>A2,>36,>0B	MONSTER D
DATA	>35,>27,>51,>61	MONSTER D
DATA	>40,>37,>21,>02,>9B,>71,>42,>00	MONSTER D
DATA	>27,>A6,>21,>A0,>51,>37,>07,>1C	MONSTER D
DATA	>E3,>DE,>84,>FE,>54,>4D,>BD,>6E	MONSTER D
DATA	>13,>72,>76,>09,>CD,>BB,>4D,>CB	MONSTER D
DATA	>D5,>39,>25,>EB	MONSTER D
DATA	>16,>21,>37,>93,>D0,>88,>53,>C4	MONSTER D
DATA	>52,>55,>39,>23,>B6,>50,>35,>F2	MONSTER D
DATA	>AB,>74,>D1,>5C,>23,>BA,>D7,>56	MONSTER D
DATA	>F9,>A5,>3A,>AB,>5D,>5D,>6A,>2B	MONSTER D
DATA	>61,>D4,>C3,>E4	MONSTER D
DATA	>AB,>B5,>86,>69,>8D,>EA,>A3,>96	MONSTER D
DATA	>9C,>6E,>51,>8A,>87,>9A,>4A,>98	MONSTER D
DATA	>69,>CA,>4B,>6A,>76,>14,>89,>95	MONSTER D
DATA	>2C,>A9,>C9,>92,>17,>65,>B2,>A2	MONSTER D
DATA	>46,>AF,>E1,>32	MONSTER D
DATA	>71,>9B,>52,>84,>7B,>E6,>3A,>AE	MONSTER D
DATA	>0A,>D3,>9B,>C9,>8E,>32,>7B,>4E	MONSTER D
DATA	>3F,>4A,>C4,>C4,>DC,>22,>60,>89	MONSTER D
DATA	>0C,>01,>FC,>B9,>26,>80,>3F,>53	MONSTER D
DATA	>10,>B0,>B9,>3B	MONSTER D
DATA	>08,>60,>98,>0B,>01,>FC,>E6,>A1	MONSTER D
DATA	>80,>BF,>42,>14,>F0,>F7,>AB,>01	MONSTER D
DATA	>FE,>DE,>34,>C0,>DF,>AF,>06,>DB	MONSTER D
DATA	>49,>A0,>19,>45,>AB,>59,>6E,>DC	MONSTER D
DATA	>64,>25,>6D,>99	MONSTER D
DATA	>B1,>71,>95,>2D,>BC,>B9,>D5,>DA	MONSTER D
DATA	>03,>00,>06,>F0,>92,>29,>01,>0A	MONSTER D
DATA	>3C,>00	MONSTER D

* MONSTER ATTACKED CREW!

DATA	#219	9
DATA	>46,>91,>D2,>B7,>24,>DD,>3B,>C5	MONSTER A
DATA	>4B,>59,>63,>F7,>95,>74,>D3,>5A	MONSTER A
DATA	>4D,>47,>71,>D6,>9B,>66,>5B,>AB	MONSTER A
DATA	>E9,>2B,>1E,>AA,>6B,>AD,>A6,>A3	MONSTER A
DATA	>7B,>2B,>2E,>A7	MONSTER A

DATA	>B9,>95,>93,>26,>45,>6D,>16,>6A	MONSTER A
DATA	>4E,>A2,>80,>00,>12,>AA,>09,>30	MONSTER A
DATA	>E3,>04,>01,>A6,>2B,>05,>07,>34	MONSTER A
DATA	>EA,>16,>00,>23,>A2,>07,>97,>83	MONSTER A
DATA	>4B,>6B,>C6,>1D	MONSTER A
DATA	>5C,>AE,>CE,>29,>59,>77,>71,>89	MONSTER A
DATA	>99,>84,>46,>9C,>C1,>9B,>64,>D1	MONSTER A
DATA	>3E,>D6,>AA,>3C,>F4,>10,>3D,>3C	MONSTER A
DATA	>CF,>C9,>32,>4A,>70,>67,>5B,>3B	MONSTER A
DATA	>C8,>9B,>A2,>54	MONSTER A
DATA	>85,>6A,>80,>03,>AA,>AE,>30,>C0	MONSTER A
DATA	>4C,>E6,>45,>ED,>41,>43,>DA,>E1	MONSTER A
DATA	>0C,>CD,>65,>CF,>6C,>76,>8C,>F4	MONSTER A
DATA	>D2,>CD,>DC,>C7,>CD,>D5,>B7,>16	MONSTER A
DATA	>0D,>04,>4C,>51	MONSTER A
DATA	>8A,>00,>2D,>5C,>09,>60,>85,>5B	MONSTER A
DATA	>00,>A6,>4B,>05,>04,>4C,>51,>0A	MONSTER A
DATA	>02,>F0,>C3,>CD,>00,>8E,>95,>2B	MONSTER A
DATA	>20,>95,>4E,>C3,>E5,>30,>16,>2B	MONSTER A
DATA	>4E,>1A,>1F,>66	MONSTER A
DATA	>70,>51,>C4,>EE,>5C,>E7,>9B,>60	MONSTER A
DATA	>A3,>6B,>7B,>CF,>E9,>52,>51,>75	MONSTER A
DATA	>B2,>2C,>DB,>8B,>C9,>30,>D9,>23	MONSTER A
DATA	>64,>AB,>46,>3B,>DA,>CE,>51,>CD	MONSTER A
DATA	>DD,>B2,>2B,>5B	MONSTER A
DATA	>47,>5A,>0F,>00	MONSTER A

* CREW MEMBER LOST

10

DATA	#153	
DATA	>04,>B0,>C7,>53,>00,>39,>57,>2A	CREW MEMB
DATA	>20,>D6,>69,>07,>A4,>D6,>59,>14	CREW MEMB
DATA	>67,>3D,>CD,>A3,>56,>57,>BA,>9C	CREW MEMB
DATA	>64,>AF,>A4,>DA,>39,>4E,>96,>92	CREW MEMB
DATA	>14,>74,>54,>3B	CREW MEMB
DATA	>C7,>C9,>72,>96,>87,>1C,>A3,>8B	CREW MEMB
DATA	>6B,>39,>DD,>CA,>50,>BC,>73,>0F	CREW MEMB
DATA	>B6,>BA,>49,>D1,>C6,>BD,>54,>2E	CREW MEMB
DATA	>47,>C5,>2B,>41,>F5,>D2,>D4,>35	CREW MEMB
DATA	>DC,>26,>C5,>54	CREW MEMB
DATA	>8B,>90,>4A,>D2,>14,>95,>22,>2A	CREW MEMB
DATA	>8A,>4E,>57,>8A,>A2,>6D,>A5,>C7	CREW MEMB
DATA	>D9,>F6,>AA,>D6,>B4,>1E,>74,>1B	CREW MEMB
DATA	>2F,>45,>5E,>C9,>34,>BA,>39,>2C	CREW MEMB
DATA	>97,>3B,>45,>56	CREW MEMB
DATA	>C9,>B4,>BA,>53,>9D,>EC,>19,>2F	CREW MEMB
DATA	>68,>5F,>23,>57,>2C,>80,>22,>32	CREW MEMB
DATA	>08,>D0,>4D,>39,>05,>10,>D0,>6D	CREW MEMB
DATA	>3B,>02,>A6,>29,>23,>C0,>54,>61	CREW MEMB
DATA	>08,>3B,>5E,>15	CREW MEMB
DATA	>40,>00,>49,>B7,>28,>A0,>51,>2F	CREW MEMB
DATA	>7B,>00	CREW MEMB

* DETECTING UNKNOWN OBJECT

11

DATA	#287	
DATA	>06,>A8,>3A,>B2,>19,>91,>BB,>79	DETECTING
DATA	>B5,>D5,>A2,>3B,>E9,>E9,>FB,>72	DETECTING
DATA	>8C,>2E,>82,>BB,>88,>CB,>00,>70	DETECTING
DATA	>40,>91,>69,>0E,>E8,>56,>BD,>A9	DETECTING
DATA	>31,>5B,>5B,>A9	DETECTING
DATA	>E2,>A1,>D4,>AA,>A6,>DD,>4E,>A7	DETECTING

DATA	>1C, >65, >24, >8B, >E0, >2E, >22, >32	DETECTING
DATA	>00, >C0, >00, >2B, >86, >34, >BE, >57	DETECTING
DATA	>63, >8B, >97, >D3, >C5, >2E, >35, >A9	DETECTING
DATA	>79, >B5, >CA, >64	DETECTING
DATA	>9A, >65, >27, >06, >23, >A7, >EC, >19	DETECTING
DATA	>92, >58, >9D, >FA, >44, >B2, >71, >4A	DETECTING
DATA	>D4, >96, >D5, >0D, >31, >E9, >D1, >85	DETECTING
DATA	>57, >B5, >E2, >A4, >A7, >98, >EE, >D9	DETECTING
DATA	>8A, >87, >9E, >43	DETECTING
DATA	>86, >56, >31, >6A, >9A, >8B, >11, >51	DETECTING
DATA	>A6, >28, >69, >C6, >7B, >7B, >9A, >6C	DETECTING
DATA	>A7, >19, >A1, >EF, >CE, >B1, >9D, >66	DETECTING
DATA	>89, >BC, >3B, >C7, >71, >5A, >E0, >F4	DETECTING
DATA	>1E, >9C, >DB, >69	DETECTING
DATA	>91, >E3, >7B, >B2, >ED, >A6, >46, >EB	DETECTING
DATA	>6A, >89, >8A, >86, >5A, >42, >38, >E7	DETECTING
DATA	>30, >19, >6A, >F6, >61, >EA, >25, >67	DETECTING
DATA	>A8, >C9, >85, >99, >95, >93, >A6, >26	DETECTING
DATA	>EB, >AE, >9E, >71	DETECTING
DATA	>86, >9A, >5C, >98, >59, >39, >69, >6A	DETECTING
DATA	>B2, >EE, >EA, >19, >27, >E8, >51, >F2	DETECTING
DATA	>78, >9B, >25, >A3, >59, >A1, >EF, >2E	DETECTING
DATA	>8E, >9C, >1E, >64, >84, >67, >59, >32	DETECTING
DATA	>C6, >B6, >E9, >29	DETECTING
DATA	>D6, >24, >99, >AE, >76, >8B, >8E, >9C	DETECTING
DATA	>4E, >4E, >A3, >1B, >C1, >80, >21, >52	DETECTING
DATA	>71, >A2, >22, >93, >8B, >EA, >52, >97	DETECTING
DATA	>82, >2D, >61, >7B, >A6, >16, >1E, >4E	DETECTING
DATA	>C0, >01, >D7, >BA	DETECTING
DATA	>37, >A7, >19, >35, >B7, >89, >D3, >6C	DETECTING
DATA	>17, >CC, >2B, >DB, >6A, >B4, >9D, >2D	DETECTING
DATA	>82, >BB, >88, >C8, >06, >02, >18, >27	DETECTING
DATA	>0B, >00, >60, >80, >A1, >C3, >18, >D0	DETECTING
DATA	>B4, >2B, >3C, >00	DETECTING
*	ZYGONAUT APPROACHING	12
DATA	#218	
DATA	>4C, >FF, >BE, >3C, >2C, >E4, >50, >5D	ZYGONAUT
DATA	>EB, >49, >B7, >D1, >5D, >D7, >D4, >36	ZYGONAUT
DATA	>CD, >D5, >4C, >5D, >55, >B2, >34, >DF	ZYGONAUT
DATA	>33, >6D, >D5, >C9, >51, >52, >CE, >D4	ZYGONAUT
DATA	>0A, >25, >4B, >C9	ZYGONAUT
DATA	>D9, >D4, >BB, >D4, >16, >B9, >26, >8C	ZYGONAUT
DATA	>DA, >56, >AC, >B4, >EC, >28, >54, >2D	ZYGONAUT
DATA	>69, >52, >5A, >26, >8D, >2A, >DB, >43	ZYGONAUT
DATA	>89, >4E, >DD, >7B, >95, >74, >35, >68	ZYGONAUT
DATA	>41, >C8, >5B, >70	ZYGONAUT
DATA	>92, >AB, >37, >23, >2A, >4B, >D7, >71	ZYGONAUT
DATA	>34, >AC, >D0, >CA, >3D, >5A, >F6, >B6	ZYGONAUT
DATA	>A6, >5D, >1D, >E3, >4C, >51, >AA, >42	ZYGONAUT
DATA	>35, >00, >34, >20, >01, >02, >44, >63	ZYGONAUT
DATA	>0C, >C5, >0E, >3E	ZYGONAUT
DATA	>3C, >6C, >E2, >1A, >4B, >04, >77, >11	ZYGONAUT
DATA	>91, >0D, >C0, >80, >92, >2A, >0C, >10	ZYGONAUT
DATA	>8A, >7B, >30, >6D, >E3, >74, >D7, >49	ZYGONAUT
DATA	>C5, >74, >59, >D3, >D3, >AF, >36, >CB	ZYGONAUT
DATA	>05, >CB, >F4, >9E	ZYGONAUT
DATA	>5B, >6C, >1B, >2D, >A3, >BA, >B6, >B1	ZYGONAUT
DATA	>44, >28, >55, >A1, >1A, >80, >00, >AA	ZYGONAUT

```

DATA >23,>05,>70,>BD,>B9,>00,>FE,>DE          ZYGONAUT
DATA >50,>C0,>CF,>CE,>46,>99,>36,>59          ZYGONAUT
DATA >35,>6B,>27,>AB                          ZYGONAUT
DATA >44,>71,>CB,>93,>9D,>EC,>1C,>29          ZYGONAUT
DATA >A2,>57,>8A,>73,>6A,>26,>CB,>28          ZYGONAUT
DATA >DA,>C2,>89,>16,>37,>C4,>34,>0B          ZYGONAUT
DATA >27,>68,>5A,>53,>71,>CD,>DC,>AB          ZYGONAUT
DATA >A9,>53,>4D,>09                          ZYGONAUT
DATA >77,>1F                                  ZYGONAUT
* CONTINUE GAME, CAPTAIN                      13
DATA #238
DATA >06,>A8,>51,>4C,>01,>59,>5A,>04          CONTINUE
DATA >35,>0B,>E7,>F2,>A5,>9D,>54,>AB          CONTINUE
DATA >2D,>AB,>1A,>A6,>D1,>94,>D0,>0F          CONTINUE
DATA >25,>55,>52,>37,>40,>75,>C1,>06          CONTINUE
DATA >38,>C3,>29,>49                          CONTINUE
DATA >35,>8B,>69,>8F,>A2,>29,>37,>69          CONTINUE
DATA >86,>34,>A6,>57,>D2,>A4,>11,>4D          CONTINUE
DATA >44,>DF,>76,>90,>9D,>B0,>B3,>60          CONTINUE
DATA >35,>5E,>4A,>52,>65,>66,>13,>C1          CONTINUE
DATA >A4,>48,>3D,>3A                          CONTINUE
DATA >84,>77,>98,>2A,>45,>45,>2A,>49          CONTINUE
DATA >EB,>49,>D4,>54,>95,>14,>6D,>46          CONTINUE
DATA >16,>CB,>56,>B2,>94,>EE,>44,>AB          CONTINUE
DATA >DE,>F1,>52,>9A,>13,>AD,>1E,>25          CONTINUE
DATA >4B,>1D,>C5,>44                          CONTINUE
DATA >6B,>14,>2F,>6D,>64,>15,>AB,>91          CONTINUE
DATA >3D,>F4,>91,>44,>BC,>47,>76,>33          CONTINUE
DATA >7A,>12,>F1,>5E,>3A,>C5,>2C,>5E          CONTINUE
DATA >34,>B2,>AC,>38,>DB,>1B,>EA,>D4          CONTINUE
DATA >70,>62,>1D,>EB                          CONTINUE
DATA >30,>5B,>04,>77,>11,>91,>0D,>00          CONTINUE
DATA >09,>C8,>DA,>C6,>01,>5D,>3B,>25          CONTINUE
DATA >7D,>A4,>D0,>B2,>12,>D3,>CC,>54          CONTINUE
DATA >3C,>23,>42,>71,>B3,>63,>8D,>F4          CONTINUE
DATA >6B,>C5,>C1,>12                          CONTINUE
DATA >C1,>5D,>44,>54,>03,>20,>20,>3A          CONTINUE
DATA >57,>30,>C0,>A0,>EE,>49,>77,>DA          CONTINUE
DATA >D3,>7B,>15,>17,>3D,>28,>4F,>CB          CONTINUE
DATA >57,>95,>74,>6F,>DC,>3D,>56,>96          CONTINUE
DATA >D3,>8C,>F6,>8E                          CONTINUE
DATA >2B,>46,>46,>33,>C2,>DE,>5C,>5C          CONTINUE
DATA >2B,>4D,>73,>7B,>4B,>71,>2D,>54          CONTINUE
DATA >21,>FC,>3D,>CD,>09,>3C                  CONTINUE
* COOLANT IS RUNNING LOW                      14
DATA #224
DATA >0A,>70,>C7,>59,>00,>6E,>98,>0B          COOLANT I
DATA >C0,>B7,>E0,>24,>EA,>54,>6C,>3A          COOLANT I
DATA >72,>9A,>E4,>44,>4F,>50,>D5,>E9          COOLANT I
DATA >72,>97,>9B,>1C,>44,>A7,>E3,>DB          COOLANT I
DATA >E9,>D2,>10,>83                          COOLANT I
DATA >2D,>8D,>58,>37,>53,>98,>7C,>E1          COOLANT I
DATA >8B,>75,>D7,>58,>05,>86,>CF,>82          COOLANT I
DATA >DE,>4A,>EC,>10,>60,>2A,>55,>05          COOLANT I
DATA >2C,>11,>C6,>94,>29,>4A,>55,>AB          COOLANT I
DATA >06,>56,>82,>92                          COOLANT I
DATA >95,>84,>D7,>AB,>A9,>5A,>32,>9C          COOLANT I
DATA >B2,>8A,>3D,>C5,>CE,>2B,>C4,>24          COOLANT I

```


DATA	>5E, >AD, >6C, >27, >4B, >04, >77, >11	WATER TAN
DATA	>91, >0D, >E0, >80, >A9, >CC, >15, >D0	WATER TAN
DATA	>6D, >5B, >52, >8B, >36, >F3, >0B, >3B	WATER TAN
DATA	>4D, >4B, >41, >23	WATER TAN
DATA	>5B, >18, >0F, >BD, >46, >57, >AF, >92	WATER TAN
DATA	>DD, >8C, >1A, >D5, >6D, >5C, >76, >33	WATER TAN
DATA	>5A, >54, >F5, >2A, >C6, >4D, >EF, >49	WATER TAN
DATA	>24, >63, >9B, >04, >BD, >04, >F0, >8B	WATER TAN
DATA	>62, >AA, >35, >A3	WATER TAN
DATA	>A5, >1A, >62, >E2, >4A, >04, >10, >8A	WATER TAN
DATA	>AA, >02, >AA, >4B, >33, >40, >37, >A9	WATER TAN
DATA	>0B, >AB, >C6, >1D, >01, >C3, >B4, >13	WATER TAN
DATA	>6D, >2A, >55, >11, >91, >81, >35, >00	WATER TAN
DATA	>04, >34, >5B, >8E	WATER TAN
DATA	>80, >A6, >53, >11, >50, >7D, >1A, >01	WATER TAN
DATA	>AA, >A9, >30, >7C, >71, >15, >CE, >A6	WATER TAN
DATA	>3A, >09, >CE, >FC, >1B, >BC, >DD, >CA	WATER TAN
DATA	>15, >DE, >BB, >6F, >94, >B3, >5A, >F9	WATER TAN
DATA	>2C, >46, >C5, >EB	WATER TAN
DATA	>CA, >94, >F5, >1D, >EB, >65, >2B, >4B	WATER TAN
DATA	>F4, >53, >B5, >4C, >FD, >00	WATER TAN

* OUT OF WATER, CAPTAIN!

DATA	#214	17
DATA	>C5, >0B, >AD, >C6, >AD, >9D, >14, >C3	OUT OF WA
DATA	>8E, >2C, >AD, >60, >DC, >F5, >A5, >C7	OUT OF WA
DATA	>D0, >A1, >B4, >4E, >A6, >9A, >D4, >6B	OUT OF WA
DATA	>22, >D4, >32, >AE, >01, >56, >F6, >70	OUT OF WA
DATA	>C0, >94, >E6, >41	OUT OF WA
DATA	>FC, >26, >C4, >CB, >12, >35, >C5, >D9	OUT OF WA
DATA	>4B, >E3, >AB, >5B, >15, >A7, >0A, >5F	OUT OF WA
DATA	>AE, >32, >71, >B4, >A6, >74, >23, >22	OUT OF WA
DATA	>5C, >DC, >89, >D3, >6B, >70, >35, >12	OUT OF WA
DATA	>25, >46, >8B, >5A	OUT OF WA
DATA	>3C, >C4, >1D, >31, >9D, >DE, >75, >93	OUT OF WA
DATA	>A9, >6D, >B5, >A2, >E9, >B2, >A1, >D0	OUT OF WA
DATA	>75, >97, >E2, >62, >55, >5B, >5B, >D9	OUT OF WA
DATA	>F2, >14, >83, >32, >9C, >5A, >BA, >39	OUT OF WA
DATA	>81, >00, >2C, >99	OUT OF WA
DATA	>AE, >80, >E9, >94, >93, >E2, >A3, >6A	OUT OF WA
DATA	>7A, >C5, >AD, >7A, >B5, >B2, >ED, >6D	OUT OF WA
DATA	>00, >05, >0C, >4D, >1D, >80, >AC, >6D	OUT OF WA
DATA	>1C, >D0, >B5, >53, >D2, >47, >0A, >2D	OUT OF WA
DATA	>2B, >31, >CD, >4C	OUT OF WA
DATA	>C5, >33, >22, >14, >37, >3B, >D6, >4B	OUT OF WA
DATA	>8F, >56, >1C, >2C, >11, >DC, >45, >44	OUT OF WA
DATA	>35, >00, >02, >A2, >73, >05, >03, >0C	OUT OF WA
DATA	>EA, >9E, >74, >A7, >3D, >BD, >57, >71	OUT OF WA
DATA	>D1, >83, >F2, >B4	OUT OF WA
DATA	>7C, >55, >49, >F7, >C6, >DD, >63, >65	OUT OF WA
DATA	>39, >CD, >6B, >EF, >8B, >62, >64, >34	OUT OF WA
DATA	>23, >EC, >CD, >C5, >B5, >D2, >34, >B7	OUT OF WA
DATA	>B7, >14, >D7, >42, >15, >C2, >DF, >D3	OUT OF WA
DATA	>9C, >3C	OUT OF WA

* CONGRATULATIONS CAPTAIN!

DATA	#263	18
DATA	>02, >8B, >9A, >43, >00, >D5, >19, >26	CONGRATUL
DATA	>25, >29, >8E, >8E, >A4, >55, >D4, >64	CONGRATUL
DATA	>B1, >3A, >93, >49, >D0, >AC, >E1, >37	CONGRATUL

DATA	>DD,>25,>4E,>E8	YOU'LL NE
DATA	>B8,>58,>52,>1F,>FB,>C8,>86,>3E	YOU'LL NE
DATA	>1C,>6E,>4C,>9E,>6A,>51,>7B,>BA	YOU'LL NE
DATA	>D3,>29,>86,>6E,>E1,>66,>9E,>D8	YOU'LL NE
DATA	>3B,>D1,>29,>6E,>18,>2E,>DA,>DC	YOU'LL NE
DATA	>B5,>3B,>DD,>71	YOU'LL NE
DATA	>8E,>AD,>6A,>F9,>9A,>5A,>28,>62	YOU'LL NE
DATA	>E8,>6C,>78,>22,>E1,>88,>16,>B9	YOU'LL NE
DATA	>A6,>4A,>92,>93,>B5,>59,>6A,>A6	YOU'LL NE
DATA	>5D,>5E,>6B,>55,>A4,>D4,>6E,>3B	YOU'LL NE
DATA	>C7,>C9,>25,>42	YOU'LL NE
DATA	>5C,>D7,>6E,>B7,>78,>AE,>6A,>68	YOU'LL NE
DATA	>D4,>62,>17,>98,>A4,>93,>55,>B6	YOU'LL NE
DATA	>4D,>68,>8B,>5A,>B6,>59,>36,>91	YOU'LL NE
DATA	>28,>5E,>99,>AA,>38,>C4,>D6,>AB	YOU'LL NE
DATA	>47,>88,>DB,>1C	YOU'LL NE
DATA	>8F,>A8,>55,>53,>B1,>58,>BB,>A3	YOU'LL NE
DATA	>1D,>EF,>78,>A7,>3B,>BB,>C5,>2A	YOU'LL NE
DATA	>7B,>56,>E5,>CE,>55,>B3,>4E,>75	YOU'LL NE
DATA	>CD,>2A,>52,>59,>33,>34,>DE,>98	YOU'LL NE
DATA	>B5,>96,>E3,>07	YOU'LL NE
END		

PAGE
GROM 6

* THIS GROM RESERVED FOR SPANISH *

```

* PASE AL NIVEL NUEVO 154
  DATA #274
  DATA >00,>4D,>35,>37,>C2,>BD,>99,>1E PASE AL N
  DATA >D5,>DD,>8C,>CC,>52,>72,>54,>BF PASE AL N
  DATA >D6,>23,>D3,>F6,>51,>DD,>AD,>88 PASE AL N
  DATA >2C,>AB,>4B,>71,>2B,>DB,>3A,>51 PASE AL N
  DATA >37,>C5,>95,>6A PASE AL N
  DATA >33,>47,>9A,>94,>1E,>D3,>4D,>5D PASE AL N
  DATA >36,>02,>7E,>74,>C3,>00,>06,>30 PASE AL N
  DATA >90,>34,>1F,>DC,>3B,>83,>F1,>D4 PASE AL N
  DATA >96,>1E,>AC,>4F,>A5,>2B,>9D,>FA PASE AL N
  DATA >D4,>9B,>5E,>BC PASE AL N
  DATA >66,>9C,>DA,>75,>7A,>F1,>9A,>F9 PASE AL N
  DATA >66,>C7,>EB,>25,>7B,>DA,>85,>53 PASE AL N
  DATA >AE,>0D,>CD,>CF,>8C,>B4,>61,>3A PASE AL N
  DATA >B5,>A1,>B9,>E1,>D5,>11,>9C,>86 PASE AL N
  DATA >E6,>62,>66,>A4 PASE AL N
  DATA >70,>4A,>BA,>72,>E9,>53,>C6,>35 PASE AL N
  DATA >EB,>9A,>87,>65,>A7,>14,>AB,>07 PASE AL N
  DATA >DD,>09,>2D,>2F,>97,>EB,>74,>A5 PASE AL N
  DATA >7D,>5D,>D5,>8D,>D7,>93,>5E,>14 PASE AL N
  DATA >C5,>34,>A9,>C9 PASE AL N
  DATA >46,>D2,>BD,>94,>99,>0D,>D9,>C6 PASE AL N
  DATA >30,>CA,>C3,>B4,>94,>4A,>B3,>18 PASE AL N
  DATA >29,>68,>C4,>27,>AB,>A9,>6F,>7D PASE AL N
  DATA >6B,>5B,>9B,>5A,>D3,>A2,>72,>A9 PASE AL N
  DATA >4C,>A5,>4E,>B3 PASE AL N
  DATA >68,>AC,>32,>EB,>5A,>55,>AB,>92 PASE AL N
  DATA >D4,>85,>17,>35,>AE,>2A,>55,>68 PASE AL N
  DATA >CF,>F6,>A2,>AA,>55,>A3,>1A,>A5 PASE AL N
  DATA >69,>E3,>E6,>BC,>96,>B4,>EO,>DD PASE AL N
  DATA >39,>CC,>6C,>D2 PASE AL N
  DATA >4D,>CE,>11,>71,>27,>59,>6F,>BA PASE AL N
  DATA >DD,>91,>92,>2D,>67,>E9,>29,>5B PASE AL N
  DATA >5A,>27,>AB,>AD,>6F,>63,>98,>21 PASE AL N
  DATA >87,>85,>2F,>1B,>67,>C9,>5E,>EC PASE AL N
  DATA >3A,>6A,>B4,>63 PASE AL N
  DATA >3C,>79,>0B,>3A,>87,>B1,>F7,>AC PASE AL N
  DATA >27,>3C,>D1,>0F,>8D,>47,>36,>F3 PASE AL N
  DATA >85,>EB,>54,>76,>D7,>0F PASE AL N
* GANO EL BONO DE PUNTOS 163
  DATA #317
  DATA >C2,>16,>CA,>2F,>24,>6C,>23,>1B GANO EL B
  DATA >DB,>E9,>56,>93,>E2,>ED,>A5,>87 GANO EL B
  DATA >18,>E5,>DE,>4C,>86,>1E,>FA,>58 GANO EL B
  DATA >65,>C5,>D9,>FA,>D1,>43,>8B,>AC GANO EL B
  DATA >18,>A5,>47,>4F GANO EL B
  DATA >31,>C2,>23,>50,>16,>3D,>58,>0E GANO EL B
  DATA >CB,>92,>55,>74,>A7,>25,>B5,>23 GANO EL B
  DATA >4C,>D5,>AB,>3E,>74,>3B,>8A,>33 GANO EL B
  DATA >92,>CD,>D4,>B7,>3E,>14,>DD,>56 GANO EL B
  DATA >54,>9F,>79,>51 GANO EL B
  DATA >D4,>5A,>88,>3C,>E9,>55,>A9,>72 GANO EL B

```

DATA	>96,>92,>A4,>EE,>88,>C4,>A3,>29	GANO EL B
DATA	>92,>7A,>C1,>5E,>C3,>3A,>48,>66	GANO EL B
DATA	>A5,>CA,>17,>13,>2C,>00,>34,>AD	GANO EL B
DATA	>04,>89,>3C,>97	GANO EL B
DATA	>B3,>B5,>A3,>47,>6D,>91,>E5,>4A	GANO EL B
DATA	>B7,>5E,>F4,>2B,>82,>FD,>8B,>5A	GANO EL B
DATA	>D5,>B3,>6E,>8D,>99,>4F,>31,>26	GANO EL B
DATA	>36,>51,>1B,>A9,>EE,>04,>DB,>CD	GANO EL B
DATA	>B6,>2E,>33,>D5	GANO EL B
DATA	>94,>24,>53,>BF,>85,>D0,>A3,>9F	GANO EL B
DATA	>CD,>6A,>54,>7D,>EB,>BE,>67,>52	GANO EL B
DATA	>BB,>99,>A2,>07,>AB,>8E,>95,>A5	GANO EL B
DATA	>92,>6E,>83,>7B,>6E,>5B,>2E,>BA	GANO EL B
DATA	>F3,>16,>9C,>1A	GANO EL B
DATA	>AD,>E9,>F6,>84,>A9,>A6,>AD,>A1	GANO EL B
DATA	>DB,>BE,>50,>D1,>B4,>BB,>1E,>74	GANO EL B
DATA	>DD,>0A,>AB,>97,>5E,>30,>B4,>6D	GANO EL B
DATA	>D1,>6C,>BA,>C5,>70,>DE,>2B,>27	GANO EL B
DATA	>14,>75,>63,>5A	GANO EL B
DATA	>D5,>42,>E0,>54,>A5,>2A,>54,>03	GANO EL B
DATA	>C0,>00,>C1,>9B,>2B,>93,>ED,>45	GANO EL B
DATA	>B3,>55,>E2,>4C,>76,>D3,>55,>DB	GANO EL B
DATA	>71,>32,>B5,>8F,>22,>B5,>BA,>D9	GANO EL B
DATA	>8C,>A6,>33,>A5	GANO EL B
DATA	>B2,>F1,>8C,>80,>DE,>B6,>36,>73	GANO EL B
DATA	>15,>F7,>9A,>80,>9B,>0B,>9C,>AA	GANO EL B
DATA	>54,>85,>6A,>80,>01,>AC,>2B,>4B	GANO EL B
DATA	>AE,>A9,>2D,>66,>8F,>26,>BB,>EA	GANO EL B
DATA	>05,>66,>0D,>62	GANO EL B
DATA	>EF,>1A,>4F,>8C,>54,>AE,>A6,>ED	GANO EL B
DATA	>C2,>AC,>C7,>9C,>54,>09,>03,>7E	GANO EL B
DATA	>74,>95,>80,>04,>3B,>40,>80,>1F	GANO EL B
DATA	>9D,>31,>80,>81,>07,>00	GANO EL B
*	MIEMBRO EXTRA TRIPULACION	173
DATA	#344	
DATA	>4A,>77,>92,>CE,>4B,>AC,>5A,>DD	MIEMBRO E
DATA	>6B,>51,>8B,>5A,>51,>13,>97,>DC	MIEMBRO E
DATA	>40,>27,>5D,>5D,>6A,>B5,>9C,>95	MIEMBRO E
DATA	>EC,>74,>2B,>5B,>99,>4A,>53,>42	MIEMBRO E
DATA	>D7,>AC,>0E,>96	MIEMBRO E
DATA	>45,>71,>92,>CE,>4B,>AC,>56,>25	MIEMBRO E
DATA	>29,>DA,>69,>F0,>46,>92,>AC,>14	MIEMBRO E
DATA	>25,>5A,>37,>63,>77,>5D,>95,>A5	MIEMBRO E
DATA	>5B,>BD,>5E,>B6,>72,>9B,>EA,>8A	MIEMBRO E
DATA	>86,>77,>48,>1A	MIEMBRO E
DATA	>AA,>49,>59,>3A,>11,>7E,>6B,>A6	MIEMBRO E
DATA	>65,>CA,>14,>9A,>A9,>75,>2D,>6B	MIEMBRO E
DATA	>5A,>C1,>0A,>52,>4A,>D4,>CC,>0D	MIEMBRO E
DATA	>35,>5B,>C1,>0A,>D5,>B6,>7E,>B5	MIEMBRO E
DATA	>A3,>39,>A7,>91	MIEMBRO E
DATA	>E9,>9C,>B7,>56,>55,>00,>01,>DB	MIEMBRO E
DATA	>6B,>E1,>80,>1A,>D2,>0D,>50,>53	MIEMBRO E
DATA	>A9,>02,>AA,>4F,>E7,>00,>02,>A7	MIEMBRO E
DATA	>2A,>55,>A1,>1A,>00,>4D,>B5,>C9	MIEMBRO E
DATA	>D2,>DB,>95,>36	MIEMBRO E
DATA	>B5,>46,>0B,>D5,>8B,>D3,>D4,>9B	MIEMBRO E
DATA	>2D,>79,>D4,>D1,>52,>5D,>CF,>2C	MIEMBRO E
DATA	>2F,>EA,>53,>ED,>6A,>50,>CC,>8C	MIEMBRO E

DATA	>74,>35,>F4,>4A,>31,>49,>35,>38	MIEMBRO E
DATA	>DB,>01,>00,>00	MIEMBRO E
DATA	>00,>40,>D1,>B4,>8F,>CC,>74,>D5	MIEMBRO E
DATA	>45,>0F,>D6,>52,>4E,>E3,>05,>B3	MIEMBRO E
DATA	>A4,>60,>92,>76,>52,>0C,>2F,>E5	MIEMBRO E
DATA	>A6,>DD,>76,>D5,>93,>6E,>A5,>6D	MIEMBRO E
DATA	>5A,>D9,>46,>BA	MIEMBRO E
DATA	>A0,>9D,>1D,>26,>1B,>40,>00,>3A	MIEMBRO E
DATA	>B5,>1B,>9D,>9F,>64,>89,>72,>12	MIEMBRO E
DATA	>74,>53,>C3,>F1,>CD,>7E,>31,>AA	MIEMBRO E
DATA	>C6,>F4,>CA,>38,>C9,>C8,>92,>36	MIEMBRO E
DATA	>62,>62,>34,>C3	MIEMBRO E
DATA	>9A,>C0,>F4,>61,>3C,>4C,>DB,>23	MIEMBRO E
DATA	>4B,>83,>5E,>37,>83,>69,>92,>87	MIEMBRO E
DATA	>9B,>A1,>0E,>D6,>08,>65,>AE,>26	MIEMBRO E
DATA	>99,>01,>3F,>BA,>52,>00,>03,>14	MIEMBRO E
DATA	>A0,>40,>D2,>B2	MIEMBRO E
DATA	>A4,>9C,>74,>46,>4D,>EF,>0E,>85	MIEMBRO E
DATA	>27,>15,>37,>BD,>18,>AA,>89,>D0	MIEMBRO E
DATA	>32,>0C,>1F,>22,>3D,>8A,>F1,>30	MIEMBRO E
DATA	>EC,>2C,>29,>59,>3A,>C3,>30,>37	MIEMBRO E
DATA	>43,>75,>68,>4F	MIEMBRO E
DATA	>B3,>5B,>C9,>D2,>C3,>9D,>4B,>A2	MIEMBRO E
DATA	>3B,>5B,>1B,>2D,>DD,>08,>65,>2D	MIEMBRO E
DATA	>6D,>72,>FB,>01	MIEMBRO E
* BUEN TIRO. CAPITAN		304
DATA	#213	
DATA	>CB,>A0,>BC,>2B,>B3,>9C,>60,>13	BUEN TIRO
DATA	>9B,>DB,>2C,>BA,>B3,>19,>29,>CE	BUEN TIRO
DATA	>36,>A9,>FC,>06,>65,>1F,>EB,>A6	BUEN TIRO
DATA	>DA,>EB,>94,>D1,>6C,>87,>1A,>53	BUEN TIRO
DATA	>BB,>C7,>72,>1A	BUEN TIRO
DATA	>6A,>4E,>9A,>D1,>C6,>7A,>AB,>39	BUEN TIRO
DATA	>69,>46,>92,>D2,>A6,>E4,>22,>91	BUEN TIRO
DATA	>41,>AA,>9B,>12,>3C,>87,>A7,>28	BUEN TIRO
DATA	>4D,>8A,>52,>BA,>11,>1A,>DF,>28	BUEN TIRO
DATA	>24,>D4,>46,>48	BUEN TIRO
DATA	>7D,>29,>33,>19,>C7,>DE,>10,>4F	BUEN TIRO
DATA	>0C,>00,>0A,>88,>D6,>BD,>28,>AB	BUEN TIRO
DATA	>10,>E5,>88,>EC,>AA,>56,>B5,>AB	BUEN TIRO
DATA	>4D,>6D,>9E,>34,>BA,>14,>35,>B3	BUEN TIRO
DATA	>A5,>72,>33,>77	BUEN TIRO
DATA	>DB,>74,>5B,>2C,>AC,>BA,>73,>30	BUEN TIRO
DATA	>C4,>33,>7E,>2F,>BA,>D1,>B4,>16	BUEN TIRO
DATA	>B1,>54,>5B,>61,>F2,>A4,>0B,>9C	BUEN TIRO
DATA	>AA,>54,>85,>6A,>00,>80,>01,>C2	BUEN TIRO
DATA	>4A,>49,>9A,>5D	BUEN TIRO
DATA	>91,>15,>26,>BB,>9B,>D5,>04,>00	BUEN TIRO
DATA	>A7,>B6,>DC,>11,>A2,>8A,>9D,>95	BUEN TIRO
DATA	>38,>FB,>A5,>2B,>D2,>16,>00,>40	BUEN TIRO
DATA	>B1,>6C,>F3,>2A,>53,>A5,>C3,>72	BUEN TIRO
DATA	>A3,>C2,>53,>19	BUEN TIRO
DATA	>4F,>6B,>9A,>DD,>EC,>66,>31,>7D	BUEN TIRO
DATA	>76,>6F,>66,>6B,>C6,>50,>56,>32	BUEN TIRO
DATA	>DA,>E9,>6A,>9D,>69,>4C,>F9,>A7	BUEN TIRO
DATA	>BB,>5D,>A4,>49,>6E,>D5,>51,>71	BUEN TIRO
DATA	>1E,>00	BUEN TIRO
* MONSTRUO DESTRUIDO		311

```

DATA #241
DATA >41,>0F,>D6,>53,>99,>A2,>54,>BD MONSTRUO
DATA >4B,>53,>1B,>AA,>F3,>95,>46,>E1 MONSTRUO
DATA >A8,>A9,>66,>AD,>04,>97,>95,>A9 MONSTRUO
DATA >4E,>75,>28,>6E,>B7,>38,>9C,>D8 MONSTRUO
DATA >A6,>DB,>12,>41 MONSTRUO
DATA >32,>72,>82,>C2,>4C,>76,>54,>84 MONSTRUO
DATA >F3,>AA,>53,>77,>0D,>55,>31,>39 MONSTRUO
DATA >0C,>FB,>D1,>4D,>02,>1A,>A0,>00 MONSTRUO
DATA >04,>20,>9B,>D0,>21,>25,>6B,>25 MONSTRUO
DATA >5B,>61,>8A,>54 MONSTRUO
DATA >8A,>AA,>99,>2B,>19,>92,>8D,>51 MONSTRUO
DATA >D2,>31,>6E,>4B,>3A,>57,>4B,>95 MONSTRUO
DATA >D5,>22,>F1,>32,>6A,>15,>8B,>B2 MONSTRUO
DATA >64,>65,>AF,>0C,>C5,>B9,>F4,>AB MONSTRUO
DATA >60,>DD,>74,>9B MONSTRUO
DATA >3C,>37,>9C,>49,>33,>54,>EB,>2A MONSTRUO
DATA >37,>A6,>C9,>B0,>DA,>3D,>C3,>EB MONSTRUO
DATA >28,>E0,>47,>57,>09,>4B,>80,>03 MONSTRUO
DATA >14,>40,>E0,>54,>A5,>2A,>54,>03 MONSTRUO
DATA >08,>50,>7D,>5A MONSTRUO
DATA >D3,>9C,>09,>CF,>1B,>D9,>45,>4B MONSTRUO
DATA >45,>4D,>2C,>D2,>38,>C3,>56,>31 MONSTRUO
DATA >D7,>4C,>12,>74,>5E,>33,>B4,>AB MONSTRUO
DATA >6C,>D4,>83,>2E,>4A,>B8,>F5,>CA MONSTRUO
DATA >4A,>BA,>57,>3C MONSTRUO
DATA >91,>C5,>2A,>E9,>D5,>42,>CD,>84 MONSTRUO
DATA >A2,>AC,>17,>3D,>4A,>AE,>DE,>54 MONSTRUO
DATA >E4,>0C,>AB,>CD,>A3,>1B,>71,>B0 MONSTRUO
DATA >7C,>12,>8D,>0A,>8D,>C1,>12,>23 MONSTRUO
DATA >7D,>3D,>D0,>3B MONSTRUO
DATA >5B,>2D,>D7,>CB,>46,>63,>6C,>B5 MONSTRUO
DATA >0A,>26,>42,>AE,>B0,>F9,>1E,>31 MONSTRUO
DATA >1E,>8B,>C2,>31,>3A,>43,>D1,>1C MONSTRUO
DATA >3F,>00 MONSTRUO

```

* MUY BIEN CAPITAN

319

```

DATA #233
DATA >42,>A1,>2E,>CF,>35,>9D,>3A,>D5 MUY BIEN
DATA >0B,>1B,>73,>A9,>1B,>D5,>AC,>66 MUY BIEN
DATA >35,>AB,>45,>95,>A6,>5A,>D5,>6D MUY BIEN
DATA >17,>45,>85,>5A,>56,>09,>5B,>64 MUY BIEN
DATA >B9,>3A,>1B,>4B MUY BIEN
DATA >55,>91,>E4,>0C,>71,>3B,>C9,>4D MUY BIEN
DATA >D4,>2E,>2C,>7C,>E8,>34,>A1,>7A MUY BIEN
DATA >B0,>EE,>64,>DA,>85,>2E,>24,>D1 MUY BIEN
DATA >6B,>0E,>AF,>56,>8C,>54,>1B,>B7 MUY BIEN
DATA >CB,>5C,>11,>D3 MUY BIEN
DATA >A8,>E9,>0C,>0B,>B9,>45,>09,>5A MUY BIEN
DATA >D2,>D5,>D5,>34,>A5,>3B,>94,>BC MUY BIEN
DATA >90,>53,>94,>A6,>DB,>7A,>59,>4A MUY BIEN
DATA >55,>BA,>BC,>A4,>EC,>AC,>B2,>DB MUY BIEN
DATA >CE,>96,>AF,>B2 MUY BIEN
DATA >95,>AD,>6E,>AD,>EB,>D6,>6B,>A6 MUY BIEN
DATA >1B,>9B,>60,>5B,>21,>17,>25,>96 MUY BIEN
DATA >8D,>E9,>0D,>63,>1B,>25,>65,>93 MUY BIEN
DATA >A6,>8C,>19,>93,>A0,>A6,>2B,>95 MUY BIEN
DATA >36,>02,>A7,>2A MUY BIEN
DATA >55,>A1,>1A,>00,>00,>0A,>90,>4D MUY BIEN

```

DATA	>A6,>E9,>E6,>46,>0E,>85,>D2,>AE	MUY BIEN
DATA	>27,>C3,>AE,>C8,>0A,>93,>0D,>00	MUY BIEN
DATA	>C9,>48,>25,>CC,>D8,>DD,>25,>C3	MUY BIEN
DATA	>49,>ED,>8D,>50	MUY BIEN
DATA	>EC,>4D,>6C,>02,>C0,>30,>5D,>8D	MUY BIEN
DATA	>68,>53,>25,>CB,>B4,>37,>3C,>9D	MUY BIEN
DATA	>15,>6F,>73,>9B,>D3,>EC,>E6,>32	MUY BIEN
DATA	>5D,>48,>2B,>56,>79,>D3,>AB,>86	MUY BIEN
DATA	>D3,>B5,>D2,>B1	MUY BIEN
DATA	>30,>7B,>4A,>A3,>2A,>CF,>53,>E3	MUY BIEN
DATA	>09,>8D,>E9,>EB,>9A,>B5,>8D,>D5	MUY BIEN
DATA	>07,>00	MUY BIEN

* LASERS RECALENTADOS

184

DATA	#277	
DATA	>48,>B7,>28,>33,>BA,>9C,>10,>9D	LASERS RE
DATA	>B9,>8E,>6D,>91,>63,>74,>69,>DD	LASERS RE
DATA	>2E,>D5,>71,>D0,>1D,>0F,>C9,>53	LASERS RE
DATA	>45,>41,>37,>3A,>64,>5A,>15,>67	LASERS RE
DATA	>AD,>6B,>47,>2B	LASERS RE
DATA	>8E,>23,>5B,>9D,>5E,>ED,>AA,>43	LASERS RE
DATA	>F5,>4A,>7F,>CA,>65,>15,>C5,>73	LASERS RE
DATA	>9E,>2A,>91,>93,>95,>A4,>D6,>2A	LASERS RE
DATA	>E0,>D5,>36,>0D,>F0,>A3,>1B,>07	LASERS RE
DATA	>2B,>40,>81,>C0	LASERS RE
DATA	>2D,>DF,>EE,>52,>76,>9A,>E8,>83	LASERS RE
DATA	>A5,>6D,>B2,>4A,>A2,>6B,>FB,>36	LASERS RE
DATA	>19,>24,>4B,>45,>0C,>1D,>C6,>D3	LASERS RE
DATA	>A5,>44,>A1,>08,>D1,>88,>65,>84	LASERS RE
DATA	>AD,>00,>BC,>10	LASERS RE
DATA	>62,>80,>9F,>D4,>38,>C0,>01,>0A	LASERS RE
DATA	>20,>70,>AA,>52,>15,>AA,>01,>08	LASERS RE
DATA	>8B,>99,>35,>A9,>25,>89,>7B,>84	LASERS RE
DATA	>93,>A0,>47,>2F,>2D,>11,>49,>BA	LASERS RE
DATA	>B6,>B4,>1C,>AC	LASERS RE
DATA	>EB,>CC,>FC,>D1,>92,>97,>F2,>4E	LASERS RE
DATA	>56,>5B,>07,>04,>F4,>98,>E6,>80	LASERS RE
DATA	>30,>53,>97,>EE,>87,>77,>F8,>48	LASERS RE
DATA	>3B,>7A,>AB,>65,>63,>6D,>6D,>E9	LASERS RE
DATA	>31,>46,>F9,>05	LASERS RE
DATA	>C7,>A6,>3B,>1B,>DA,>99,>6C,>BB	LASERS RE
DATA	>D1,>8D,>6D,>0C,>C3,>5A,>F3,>4C	LASERS RE
DATA	>65,>52,>0D,>67,>0B,>E3,>17,>C9	LASERS RE
DATA	>52,>AD,>21,>0C,>1A,>EA,>43,>45	LASERS RE
DATA	>0E,>31,>B2,>12	LASERS RE
DATA	>1B,>F3,>C6,>60,>00,>13,>BD,>86	LASERS RE
DATA	>6E,>5F,>7A,>8E,>29,>D9,>FA,>D5	LASERS RE
DATA	>B7,>B1,>8D,>69,>34,>53,>95,>72	LASERS RE
DATA	>AD,>C1,>5C,>AD,>60,>C9,>9D,>D0	LASERS RE
DATA	>B5,>48,>BD,>6D	LASERS RE
DATA	>1C,>BE,>93,>A2,>92,>32,>77,>95	LASERS RE
DATA	>E7,>F4,>B0,>89,>3A,>23,>C0,>8F	LASERS RE
DATA	>6E,>1C,>90,>00,>07,>3B,>40,>01	LASERS RE
DATA	>7B,>00	LASERS RE

* MONSTRUO HA DESTRUIDO BUGUE

193

DATA	#343	
DATA	>45,>D5,>C6,>4A,>59,>AA,>0F,>D5	MONSTRUO
DATA	>AA,>2B,>13,>49,>3C,>54,>9D,>D6	MONSTRUO
DATA	>13,>C3,>CE,>54,>A7,>32,>14,>53	MONSTRUO

```

DATA >9B,>93,>9A,>6A,>97,>8B,>AC,>62      MONSTRUO
DATA >98,>7A,>84,>0F      MONSTRUO
DATA >92,>30,>9E,>D1,>15,>26,>48,>41      MONSTRUO
DATA >8B,>87,>64,>62,>01,>FC,>E8,>AA      MONSTRUO
DATA >01,>09,>20,>70,>AA,>52,>15,>AA      MONSTRUO
DATA >01,>90,>C4,>96,>72,>CA,>5C,>D1      MONSTRUO
DATA >12,>13,>1B,>4B      MONSTRUO
DATA >6B,>2B,>43,>CC,>4D,>D4,>D5,>ED      MONSTRUO
DATA >36,>C1,>66,>1F,>6D,>0F,>D3,>04      MONSTRUO
DATA >DD,>AA,>B1,>D3,>6A,>11,>D4,>33      MONSTRUO
DATA >B5,>4B,>C6,>55,>CB,>82,>12,>CC      MONSTRUO
DATA >98,>44,>5F,>B7      MONSTRUO
DATA >48,>EB,>26,>AB,>DC,>24,>25,>C5      MONSTRUO
DATA >BD,>AC,>28,>97,>57,>B5,>AA,>4F      MONSTRUO
DATA >7D,>EA,>53,>9B,>5A,>D7,>8A,>E6      MONSTRUO
DATA >AA,>65,>BA,>61,>AE,>1A,>03,>4C      MONSTRUO
DATA >0C,>03,>64,>10      MONSTRUO
DATA >55,>D5,>91,>82,>B1,>19,>AC,>1E      MONSTRUO
DATA >CA,>D3,>42,>76,>53,>8D,>F7,>1C      MONSTRUO
DATA >53,>2D,>5D,>EF,>7A,>D1,>93,>31      MONSTRUO
DATA >76,>0F,>C5,>0E,>28,>A9,>DD,>03      MONSTRUO
DATA >0E,>28,>39,>DD      MONSTRUO
DATA >00,>2D,>54,>10,>60,>9A,>54,>00      MONSTRUO
DATA >07,>38,>1B,>5A,>54,>1D,>BC,>74      MONSTRUO
DATA >C3,>B6,>D1,>6D,>74,>13,>13,>37      MONSTRUO
DATA >49,>57,>DE,>CB,>2B,>65,>55,>B5      MONSTRUO
DATA >A9,>51,>4A,>F5      MONSTRUO
DATA >18,>ED,>AE,>76,>35,>28,>56,>9B      MONSTRUO
DATA >77,>07,>1C,>A7,>05,>13,>24,>1E      MONSTRUO
DATA >6E,>87,>AC,>D3,>B0,>C7,>B1,>EE      MONSTRUO
DATA >4A,>57,>92,>AC,>6E,>62,>D6,>31      MONSTRUO
DATA >09,>92,>AC,>0F      MONSTRUO
DATA >93,>8B,>DB,>CB,>EC,>1D,>B4,>27      MONSTRUO
DATA >6D,>A4,>21,>7B,>9D,>92,>AA,>91      MONSTRUO
DATA >16,>BB,>A9,>69,>A4,>45,>1A,>34      MONSTRUO
DATA >DB,>ED,>2E,>1B,>9B,>C1,>0A,>D1      MONSTRUO
DATA >C3,>D4,>14,>3B      MONSTRUO
DATA >9D,>BD,>A4,>AB,>67,>E5,>34,>FA      MONSTRUO
DATA >16,>AA,>96,>72,>D0,>C4,>13,>AB      MONSTRUO
DATA >79,>3A,>5E,>57,>86,>EB,>0D,>A2      MONSTRUO
DATA >E9,>04,>00,>1C,>90,>93,>7A,>D1      MONSTRUO
DATA >A3,>96,>EB,>64      MONSTRUO
DATA >36,>D3,>EC,>56,>4D,>9D,>95,>14      MONSTRUO
DATA >79,>B5,>B2,>55,>EE,>4A,>A6,>E1      MONSTRUO
DATA >0E,>FB,>01,>00      MONSTRUO
* MONSTRUO HA ATACADO TRIPULACION      204
DATA #441
DATA >46,>E5,>36,>C6,>CC,>EB,>27,>CD      MONSTRUO
DATA >1A,>4F,>11,>09,>5F,>D5,>AE,>76      MONSTRUO
DATA >75,>AB,>C6,>56,>B9,>BB,>A3,>A6      MONSTRUO
DATA >AA,>7A,>96,>9B,>96,>A7,>3A,>14      MONSTRUO
DATA >D7,>16,>4A,>96      MONSTRUO
DATA >CA,>54,>86,>6C,>BA,>71,>4A,>D1      MONSTRUO
DATA >29,>92,>B0,>91,>D1,>19,>36,>4B      MONSTRUO
DATA >49,>72,>D2,>3D,>2D,>1C,>19,>20      MONSTRUO
DATA >4B,>95,>00,>74,>5F,>1A,>80,>6A      MONSTRUO
DATA >42,>05,>90,>8B      MONSTRUO
DATA >0A,>40,>02,>AA,>75,>5B,>A2,>33      MONSTRUO

```

DATA	>BE, >5E, >6F, >BB, >28, >A1, >48, >AA	MONSTRUO
DATA	>5B, >EA, >25, >E9, >96, >CD, >15, >76	MONSTRUO
DATA	>86, >DB, >44, >75, >8B, >32, >5A, >7C	MONSTRUO
DATA	>17, >9A, >A0, >6E	MONSTRUO
DATA	>AA, >D4, >33, >CE, >A2, >00, >D4, >63	MONSTRUO
DATA	>43, >80, >79, >A2, >00, >00, >00, >00	MONSTRUO
DATA	>60, >00, >B7, >AB, >BF, >6C, >EF, >44	MONSTRUO
DATA	>78, >5A, >BD, >DA, >D5, >AE, >76, >B5	MONSTRUO
DATA	>AB, >1D, >D5, >ED	MONSTRUO
DATA	>2C, >6F, >97, >7B, >D5, >AB, >2C, >C5	MONSTRUO
DATA	>8D, >4A, >EB, >65, >BD, >0B, >07, >98	MONSTRUO
DATA	>53, >A9, >00, >B7, >D2, >4E, >6E, >6F	MONSTRUO
DATA	>44, >9B, >2A, >BD, >D9, >D6, >9B, >61	MONSTRUO
DATA	>4A, >7B, >9A, >A3	MONSTRUO
DATA	>06, >80, >A5, >05, >57, >91, >1E, >74	MONSTRUO
DATA	>96, >EE, >86, >67, >8B, >C9, >9B, >86	MONSTRUO
DATA	>32, >9C, >0F, >0E, >0F, >AA, >00, >60	MONSTRUO
DATA	>80, >30, >CD, >86, >E6, >76, >45, >B4	MONSTRUO
DATA	>DA, >BD, >FA, >D5	MONSTRUO
DATA	>96, >66, >6A, >8A, >75, >63, >99, >5A	MONSTRUO
DATA	>57, >BB, >DA, >D5, >A4, >88, >D9, >DB	MONSTRUO
DATA	>99, >AB, >BD, >A2, >14, >D9, >02, >A2	MONSTRUO
DATA	>93, >0E, >00, >02, >DC, >CB, >40, >80	MONSTRUO
DATA	>4D, >A9, >0B, >70	MONSTRUO
DATA	>CD, >15, >01, >BE, >59, >20, >C0, >27	MONSTRUO
DATA	>65, >00, >00, >00, >0B, >40, >70, >1E	MONSTRUO
DATA	>43, >D5, >36, >36, >26, >94, >15, >BD	MONSTRUO
DATA	>3A, >76, >AE, >B0, >32, >34, >AD, >E2	MONSTRUO
DATA	>A6, >4A, >76, >D7	MONSTRUO
DATA	>82, >CE, >63, >8E, >57, >D2, >06, >00	MONSTRUO
DATA	>05, >EB, >D0, >E6, >42, >7E, >55, >6C	MONSTRUO
DATA	>4A, >75, >D2, >4D, >F6, >C0, >17, >7B	MONSTRUO
DATA	>C5, >C8, >1A, >3B, >CA, >13, >15, >23	MONSTRUO
DATA	>2B, >EA, >B2, >4D	MONSTRUO
DATA	>D0, >0C, >AF, >AD, >CD, >47, >C9, >32	MONSTRUO
DATA	>5D, >F3, >CE, >0B, >36, >C3, >B4, >3D	MONSTRUO
DATA	>83, >24, >A4, >27, >D3, >54, >8B, >12	MONSTRUO
DATA	>63, >12, >EC, >61, >8B, >8D, >5D, >B6	MONSTRUO
DATA	>02, >86, >2B, >E3	MONSTRUO
DATA	>00, >01, >96, >29, >63, >40, >0F, >A9	MONSTRUO
DATA	>02, >AB, >AB, >39, >68, >51, >70, >6E	MONSTRUO
DATA	>39, >93, >A4, >37, >07, >B1, >D5, >AC	MONSTRUO
DATA	>AA, >DE, >8C, >AB, >A5, >3D, >D2, >53	MONSTRUO
DATA	>33, >6C, >4F, >0D	MONSTRUO
DATA	>5D, >26, >CD, >30, >AD, >DA, >70, >14	MONSTRUO
DATA	>4D, >B3, >9B, >C5, >D2, >C3, >92, >3D	MONSTRUO
DATA	>95, >05, >5B, >46, >4B, >ED, >B0, >A3	MONSTRUO
DATA	>6C, >AE, >7D, >2B, >42, >96, >B4, >8B	MONSTRUO
DATA	>25, >B4, >86, >E7	MONSTRUO
DATA	>3B, >22, >76, >B4, >6C, >2A, >9D, >EB	MONSTRUO
DATA	>01, >00	MONSTRUO
*	PERDIDO HOMBRE TRIPULACION	218
DATA	#346	
DATA	>00, >4B, >49, >49, >33, >3B, >59, >5D	PERDIDO H
DATA	>F5, >AA, >57, >5D, >6A, >CA, >9A, >BA	PERDIDO H
DATA	>62, >67, >AB, >3E, >68, >5B, >27, >AF	PERDIDO H
DATA	>A9, >06, >55, >05, >B7, >8C, >64, >AD	PERDIDO H
DATA	>34, >45, >CB, >3C	PERDIDO H

DATA	>5A, >2E, >D3, >98, >4E, >CB, >6A, >B9	PERDIDO H
DATA	>4E, >73, >42, >35, >C3, >95, >16, >35	PERDIDO H
DATA	>0B, >AA, >2E, >A1, >DD, >95, >AE, >74	PERDIDO H
DATA	>A5, >CB, >45, >F6, >4E, >AC, >46, >69	PERDIDO H
DATA	>3B, >59, >FB, >2A	PERDIDO H
DATA	>9F, >64, >DC, >E5, >25, >EB, >DE, >E6	PERDIDO H
DATA	>31, >6B, >B7, >BC, >E5, >26, >A9, >17	PERDIDO H
DATA	>D0, >51, >98, >AB, >98, >44, >35, >17	PERDIDO H
DATA	>9A, >47, >2D, >40, >53, >F4, >1C, >17	PERDIDO H
DATA	>39, >A7, >4B, >D5	PERDIDO H
DATA	>67, >82, >A5, >BD, >1C, >C5, >D4, >4D	PERDIDO H
DATA	>C7, >D4, >7A, >14, >33, >DA, >03, >CB	PERDIDO H
DATA	>F6, >55, >B4, >B2, >54, >EB, >CB, >13	PERDIDO H
DATA	>C3, >76, >51, >A5, >F5, >66, >D7, >A6	PERDIDO H
DATA	>55, >A9, >4A, >56	PERDIDO H
DATA	>A2, >32, >14, >99, >32, >3A, >96, >F5	PERDIDO H
DATA	>50, >6C, >92, >AA, >77, >3B, >4B, >0D	PERDIDO H
DATA	>56, >23, >AB, >14, >1D, >35, >5B, >ED	PERDIDO H
DATA	>1C, >57, >7C, >D5, >AD, >6E, >35, >A9	PERDIDO H
DATA	>32, >C5, >8C, >1A	PERDIDO H
DATA	>2D, >00, >00, >6B, >5A, >CD, >65, >9E	PERDIDO H
DATA	>21, >B9, >6B, >4A, >57, >74, >86, >D2	PERDIDO H
DATA	>A2, >25, >6E, >5A, >51, >4A, >8A, >66	PERDIDO H
DATA	>95, >7E, >B7, >CB, >AE, >3A, >33, >B0	PERDIDO H
DATA	>9F, >0B, >0E, >A7	PERDIDO H
DATA	>CB, >D0, >2C, >B2, >3C, >ED, >80, >01	PERDIDO H
DATA	>42, >4B, >77, >BA, >BB, >05, >1B, >AB	PERDIDO H
DATA	>3A, >EA, >C5, >2C, >B1, >C5, >DA, >92	PERDIDO H
DATA	>16, >23, >6B, >9C, >8C, >4C, >D9, >8C	PERDIDO H
DATA	>24, >A4, >B7, >D7	PERDIDO H
DATA	>E5, >32, >6C, >4A, >2F, >71, >64, >DB	PERDIDO H
DATA	>AC, >66, >B0, >9A, >49, >15, >0E, >D9	PERDIDO H
DATA	>04, >F8, >D1, >B9, >03, >14, >A0, >80	PERDIDO H
DATA	>04, >B2, >16, >84, >F8, >A4, >B3, >2D	PERDIDO H
DATA	>5A, >75, >18, >33	PERDIDO H
DATA	>CD, >BA, >EB, >4D, >D3, >DE, >2B, >7C	PERDIDO H
DATA	>55, >15, >5D, >6D, >E3, >D1, >62, >5B	PERDIDO H
DATA	>0C, >7D, >4B, >8D, >0F, >4E, >35, >8B	PERDIDO H
DATA	>A5, >6E, >B0, >73, >2B, >4F, >B6, >5C	PERDIDO H
DATA	>EA, >12, >A9, >36	PERDIDO H
DATA	>DB, >C2, >DA, >7A, >96, >6C, >6F, >6B	PERDIDO H
DATA	>8B, >59, >42, >6B, >7B, >B5, >23, >E6	PERDIDO H
DATA	>7B, >A9, >66, >DE, >8E, >1E	PERDIDO H
*	OBJETO DESCONOCIDO DELANTE	229
DATA	#361	
DATA	>46, >0B, >FA, >66, >60, >9D, >35, >CD	OBJETO DE
DATA	>BA, >37, >C5, >B3, >D6, >B5, >AE, >25	OBJETO DE
DATA	>4D, >8E, >66, >B3, >67, >E2, >35, >A2	OBJETO DE
DATA	>F3, >D0, >2C, >5A, >49, >94, >6E, >9A	OBJETO DE
DATA	>31, >F9, >CB, >71	OBJETO DE
DATA	>46, >CA, >45, >16, >CE, >5B, >01, >D9	OBJETO DE
DATA	>9B, >0A, >A0, >25, >37, >01, >34, >17	OBJETO DE
DATA	>69, >80, >91, >DD, >1C, >D0, >54, >54	OBJETO DE
DATA	>53, >A2, >B5, >6E, >57, >24, >4B, >4D	OBJETO DE
DATA	>C9, >2D, >7A, >18	OBJETO DE
DATA	>5D, >75, >AB, >49, >8D, >D1, >59, >33	OBJETO DE
DATA	>AD, >20, >70, >AA, >52, >15, >AA, >01	OBJETO DE
DATA	>00, >02, >E0, >33, >34, >C9, >32, >9F	OBJETO DE

DATA	>7B,>96,>FC,>2C,>47,>C9,>4B,>44	OBJETO DE
DATA	>A1,>6E,>3D,>CD	OBJETO DE
DATA	>65,>43,>50,>8A,>0E,>33,>4B,>C7	OBJETO DE
DATA	>45,>89,>5E,>3D,>B3,>98,>74,>BD	OBJETO DE
DATA	>EB,>41,>B7,>CA,>DB,>42,>E9,>05	OBJETO DE
DATA	>B3,>87,>54,>95,>90,>4D,>80,>1F	OBJETO DE
DATA	>DD,>31,>80,>01	OBJETO DE
DATA	>06,>A4,>50,>0E,>C0,>00,>FD,>5C	OBJETO DE
DATA	>8A,>66,>7A,>51,>DA,>32,>CD,>7A	OBJETO DE
DATA	>D2,>9D,>B3,>E2,>50,>C7,>51,>8F	OBJETO DE
DATA	>7A,>30,>F4,>48,>2E,>6A,>24,>DE	OBJETO DE
DATA	>70,>46,>E4,>A9	OBJETO DE
DATA	>4E,>E6,>B4,>01,>3F,>BA,>51,>00	OBJETO DE
DATA	>03,>1B,>C0,>00,>06,>92,>DA,>2C	OBJETO DE
DATA	>79,>6F,>C9,>AA,>5A,>D5,>AA,>16	OBJETO DE
DATA	>B4,>A0,>D4,>A3,>8E,>B6,>D1,>8D	OBJETO DE
DATA	>4B,>F3,>3C,>A6	OBJETO DE
DATA	>59,>CD,>6A,>50,>55,>5B,>0B,>1F	OBJETO DE
DATA	>7A,>51,>B5,>AA,>50,>E4,>28,>98	OBJETO DE
DATA	>0A,>FB,>44,>E5,>B6,>C3,>35,>9D	OBJETO DE
DATA	>00,>00,>51,>AB,>5B,>4F,>73,>D9	OBJETO DE
DATA	>10,>94,>A2,>C3	OBJETO DE
DATA	>CC,>D2,>71,>51,>A2,>57,>CF,>2C	OBJETO DE
DATA	>26,>4B,>2F,>CE,>BC,>72,>19,>6F	OBJETO DE
DATA	>63,>1B,>DD,>48,>86,>36,>CD,>DB	OBJETO DE
DATA	>EE,>2C,>1B,>D9,>58,>86,>6F,>19	OBJETO DE
DATA	>95,>6E,>6F,>19	OBJETO DE
DATA	>6E,>74,>FA,>BA,>DC,>6D,>0C,>23	OBJETO DE
DATA	>AC,>B2,>2C,>55,>DD,>0C,>DF,>52	OBJETO DE
DATA	>D7,>34,>4A,>32,>5C,>56,>2B,>0D	OBJETO DE
DATA	>24,>CE,>30,>4A,>DB,>5A,>AD,>7B	OBJETO DE
DATA	>43,>19,>4A,>DA	OBJETO DE
DATA	>7A,>8A,>2A,>62,>6A,>EA,>99,>D6	OBJETO DE
DATA	>4E,>01,>92,>99,>9B,>63,>7B,>DA	OBJETO DE
DATA	>4D,>96,>D7,>D6,>97,>C3,>3A,>DB	OBJETO DE
DATA	>CE,>B1,>2A,>EE,>CD,>11,>2B,>DB	OBJETO DE
DATA	>29,>ED,>09,>86	OBJETO DE
DATA	>FC,>00	OBJETO DE

* SE ACERCA EL ZYGONAUT

DATA	#276	241
DATA	>0C,>FB,>D1,>95,>03,>1C,>1B,>AA	SE ACERCA
DATA	>73,>16,>71,>A1,>F4,>AB,>39,>B9	SE ACERCA
DATA	>65,>B7,>96,>AB,>1E,>CD,>8E,>8C	SE ACERCA
DATA	>B2,>66,>7A,>B5,>AD,>15,>3D,>DA	SE ACERCA
DATA	>4C,>27,>A7,>9A	SE ACERCA
DATA	>75,>06,>FC,>EB,>CE,>01,>0A,>50	SE ACERCA
DATA	>A0,>49,>26,>4B,>F5,>98,>EA,>A3	SE ACERCA
DATA	>A4,>6A,>51,>3D,>AC,>AF,>72,>95	SE ACERCA
DATA	>AB,>0C,>25,>25,>4F,>BA,>94,>D3	SE ACERCA
DATA	>95,>AA,>20,>70	SE ACERCA
DATA	>AA,>52,>15,>AA,>01,>60,>80,>D0	SE ACERCA
DATA	>5C,>96,>E4,>A7,>65,>45,>C9,>3D	SE ACERCA
DATA	>B2,>DB,>19,>9D,>6A,>F7,>CA,>5B	SE ACERCA
DATA	>1E,>4A,>48,>91,>39,>CB,>A6,>6A	SE ACERCA
DATA	>00,>60,>00,>ED	SE ACERCA
DATA	>E2,>96,>96,>BC,>66,>8D,>C9,>9E	SE ACERCA
DATA	>5A,>D2,>82,>72,>CA,>33,>C5,>49	SE ACERCA
DATA	>0F,>D2,>2C,>AB,>E5,>3A,>DD,>CB	SE ACERCA

DATA	>CE, >92, >32, >BF, >4C, >ED, >38, >D3	CONTINUE
DATA	>73, >E9, >4D, >B5, >4D, >EC, >E4, >88	CONTINUE
DATA	>2B, >3B, >31, >00, >34, >DD, >A5, >88	CONTINUE
DATA	>0C, >67, >33, >4C, >7B, >23, >D3, >55	CONTINUE
DATA	>D1, >34, >BB, >D9	CONTINUE
DATA	>CD, >61, >BA, >E4, >91, >E4, >D2, >BB	CONTINUE
DATA	>99, >0C, >9B, >35, >8B, >02, >AD, >D1	CONTINUE
DATA	>95, >B4, >B1, >B0, >EB, >56, >13, >1A	CONTINUE
DATA	>31, >B9, >D9, >2E, >8D, >69, >5A, >72	CONTINUE
DATA	>45, >64, >F4, >07	CONTINUE
*	BAJO NIVEL DE AGUA	260
DATA	#282	
DATA	>CB, >A4, >BC, >AB, >2B, >EC, >60, >0B	BAJO NIVE
DATA	>5B, >DB, >1C, >7A, >B0, >ED, >6E, >EE	BAJO NIVE
DATA	>E4, >EB, >76, >55, >46, >AC, >92, >A3	BAJO NIVE
DATA	>BB, >67, >1D, >69, >4A, >AE, >BE, >B5	BAJO NIVE
DATA	>A1, >B9, >55, >EE	BAJO NIVE
DATA	>12, >68, >9A, >16, >D2, >78, >4A, >40	BAJO NIVE
DATA	>51, >B0, >DD, >C5, >04, >D0, >AB, >81	BAJO NIVE
DATA	>00, >72, >17, >20, >80, >F7, >00, >25	BAJO NIVE
DATA	>C9, >6F, >D4, >51, >BB, >BB, >A4, >9F	BAJO NIVE
DATA	>71, >F7, >62, >AA	BAJO NIVE
DATA	>52, >91, >54, >0F, >B7, >57, >A5, >4D	BAJO NIVE
DATA	>F2, >D9, >4D, >8F, >C5, >3A, >99, >5B	BAJO NIVE
DATA	>6F, >6F, >B5, >EE, >15, >A7, >0B, >ED	BAJO NIVE
DATA	>6F, >A6, >CE, >92, >9A, >05, >57, >39	BAJO NIVE
DATA	>39, >4E, >7A, >55	BAJO NIVE
DATA	>E0, >D3, >AC, >24, >EA, >41, >8F, >92	BAJO NIVE
DATA	>BA, >D3, >14, >2B, >5D, >CB, >74, >D3	BAJO NIVE
DATA	>74, >A2, >F5, >A6, >7B, >E3, >5D, >65	BAJO NIVE
DATA	>9A, >A6, >B1, >8D, >6D, >6C, >63, >E9	BAJO NIVE
DATA	>C1, >9B, >6E, >3B	BAJO NIVE
DATA	>EB, >AD, >77, >3D, >EB, >D6, >3A, >7D	BAJO NIVE
DATA	>B1, >AA, >AB, >59, >4D, >69, >86, >87	BAJO NIVE
DATA	>C5, >26, >3C, >A1, >25, >46, >11, >DE	BAJO NIVE
DATA	>B1, >91, >C6, >59, >66, >45, >24, >09	BAJO NIVE
DATA	>5A, >8D, >69, >9E	BAJO NIVE
DATA	>22, >A7, >E9, >3E, >48, >8E, >8B, >EA	BAJO NIVE
DATA	>66, >58, >1B, >5B, >25, >AA, >BB, >D9	BAJO NIVE
DATA	>CC, >18, >AD, >62, >54, >F1, >34, >B7	BAJO NIVE
DATA	>39, >4C, >77, >23, >2B, >D5, >C9, >36	BAJO NIVE
DATA	>B7, >39, >4C, >F3	BAJO NIVE
DATA	>3C, >33, >82, >CD, >34, >A7, >59, >2C	BAJO NIVE
DATA	>FD, >9C, >2B, >0A, >8D, >B1, >F9, >53	BAJO NIVE
DATA	>9C, >2A, >DA, >C4, >A6, >E7, >D3, >48	BAJO NIVE
DATA	>E5, >20, >5B, >F1, >E9, >30, >93, >65	BAJO NIVE
DATA	>6C, >79, >83, >33	BAJO NIVE
DATA	>8E, >5A, >B4, >8B, >6D, >6E, >6A, >BB	BAJO NIVE
DATA	>CB, >CE, >B6, >B3, >4D, >AF, >74, >0A	BAJO NIVE
DATA	>D1, >C6, >76, >35, >39, >50, >E9, >2A	BAJO NIVE
DATA	>3B, >DB, >97, >16, >A7, >FD	BAJO NIVE
*	AGUA DETECTADA	269
DATA	#177	
DATA	>45, >77, >77, >2D, >2A, >1C, >4D, >7D	AGUA DETE
DATA	>E9, >7E, >B6, >67, >95, >BD, >AD, >4F	AGUA DETE
DATA	>AD, >69, >66, >A6, >7B, >35, >E6, >A9	AGUA DETE
DATA	>39, >55, >3C, >E5, >EC, >66, >25, >54	AGUA DETE
DATA	>F2, >4A, >44, >C2	AGUA DETE

DATA	>11,>51,>E9,>2F,>52,>49,>DB,>4A	AGUA DETE
DATA	>61,>DF,>C4,>BB,>15,>0E,>C5,>DC	AGUA DETE
DATA	>D0,>CA,>43,>3A,>95,>A3,>D8,>5E	AGUA DETE
DATA	>D1,>1E,>EC,>B6,>32,>14,>5B,>23	AGUA DETE
DATA	>7B,>95,>93,>51	AGUA DETE
DATA	>B4,>09,>B7,>31,>24,>46,>49,>56	AGUA DETE
DATA	>55,>55,>9D,>16,>55,>C5,>AB,>0E	AGUA DETE
DATA	>61,>D3,>B5,>A9,>57,>03,>00,>8A	AGUA DETE
DATA	>51,>BA,>AB,>45,>48,>9B,>46,>31	AGUA DETE
DATA	>AC,>D5,>EB,>54	AGUA DETE
DATA	>46,>D6,>40,>E0,>54,>A5,>2A,>54	AGUA DETE
DATA	>03,>00,>00,>14,>10,>92,>44,>33	AGUA DETE
DATA	>EC,>F3,>CA,>65,>BB,>D3,>9B,>C6	AGUA DETE
DATA	>34,>86,>61,>5E,>79,>27,>DB,>99	AGUA DETE
DATA	>C6,>30,>55,>F3	AGUA DETE
DATA	>88,>52,>4E,>D9,>2A,>96,>D9,>19	AGUA DETE
DATA	>E5,>CE,>B4,>DB,>DD,>2E,>B6,>5D	AGUA DETE
DATA	>ED,>E9,>62,>37,>D8,>2E,>55,>04	AGUA DETE
DATA	>1B,>7D,>E5,>84,>92,>E6,>2A,>80	AGUA DETE
DATA	>1F,>00	AGUA DETE
*	DEPOSITOS DE AGUA LLENOS	275
DATA	#360	
DATA	>4B,>57,>34,>AC,>BC,>D2,>22,>DD	DEPOSITOS
DATA	>80,>C8,>89,>4B,>84,>74,>22,>AB	DEPOSITOS
DATA	>2B,>CA,>0E,>D2,>35,>8A,>AC,>2C	DEPOSITOS
DATA	>3B,>43,>2B,>DD,>4D,>2D,>9C,>1C	DEPOSITOS
DATA	>AD,>26,>0D,>5F	DEPOSITOS
DATA	>53,>7C,>B5,>AB,>35,>CD,>47,>0B	DEPOSITOS
DATA	>2B,>57,>83,>C0,>A9,>4A,>55,>AB	DEPOSITOS
DATA	>06,>80,>03,>6C,>4E,>2F,>AA,>BA	DEPOSITOS
DATA	>4D,>6A,>CF,>A6,>AA,>43,>35,>AB	DEPOSITOS
DATA	>CB,>63,>E0,>75	DEPOSITOS
DATA	>A5,>28,>4E,>87,>BB,>BB,>63,>05	DEPOSITOS
DATA	>FC,>EB,>42,>01,>0A,>3B,>69,>94	DEPOSITOS
DATA	>31,>D5,>B2,>55,>E4,>6C,>51,>6B	DEPOSITOS
DATA	>DD,>59,>95,>91,>22,>44,>67,>97	DEPOSITOS
DATA	>A5,>06,>30,>40	DEPOSITOS
DATA	>0C,>91,>43,>51,>A5,>34,>2B,>D5	DEPOSITOS
DATA	>0C,>C5,>8C,>16,>8D,>63,>D5,>D5	DEPOSITOS
DATA	>A2,>9A,>6B,>5B,>59,>6A,>92,>26	DEPOSITOS
DATA	>7B,>B2,>67,>A9,>49,>BA,>D7,>E5	DEPOSITOS
DATA	>22,>E1,>C6,>19	DEPOSITOS
DATA	>43,>29,>AB,>9B,>1B,>01,>FC,>EB	DEPOSITOS
DATA	>8E,>01,>67,>9B,>E0,>1C,>91,>8A	DEPOSITOS
DATA	>87,>11,>0B,>7B,>95,>AB,>5E,>46	DEPOSITOS
DATA	>80,>36,>8D,>CE,>BB,>9A,>CE,>EC	DEPOSITOS
DATA	>C5,>CC,>FA,>1D	DEPOSITOS
DATA	>07,>B3,>44,>8F,>8C,>51,>C4,>4C	DEPOSITOS
DATA	>3D,>7F,>5A,>53,>E9,>32,>FD,>EC	DEPOSITOS
DATA	>8B,>0E,>27,>D7,>BC,>C6,>D5,>AF	DEPOSITOS
DATA	>B6,>34,>73,>42,>22,>07,>CD,>D4	DEPOSITOS
DATA	>82,>A6,>6E,>43	DEPOSITOS
DATA	>DB,>60,>26,>3A,>B9,>67,>46,>EA	DEPOSITOS
DATA	>98,>EB,>41,>61,>AF,>31,>FC,>5B	DEPOSITOS
DATA	>37,>45,>ED,>A0,>8A,>61,>76,>14	DEPOSITOS
DATA	>D7,>33,>53,>8B,>FA,>56,>A6,>92	DEPOSITOS
DATA	>14,>9F,>13,>53	DEPOSITOS
DATA	>9C,>AD,>53,>62,>72,>35,>11,>A6	DEPOSITOS

DATA	>DA, >35, >4E, >8A, >BD, >8C, >30, >93	NO ME ALC
DATA	>2D, >C5, >B7, >2E, >C6, >D4, >D6, >06	NO ME ALC
DATA	>47, >E6, >1E, >F6	NO ME ALC
DATA	>5C, >19, >4C, >95, >BB, >D1, >FC, >96	NO ME ALC
DATA	>33, >78, >99, >C2, >B2, >26, >C6, >50	NO ME ALC
DATA	>35, >5A, >D9, >E2, >24, >43, >4D, >0B	NO ME ALC
DATA	>B5, >48, >53, >0C, >37, >D5, >D4, >B3	NO ME ALC
DATA	>D1, >30, >43, >31	NO ME ALC
DATA	>37, >CF, >45, >CB, >0C, >29, >42, >23	NO ME ALC
DATA	>E6, >0E, >2B, >FA, >68, >35, >7B, >3A	NO ME ALC
DATA	>2C, >EF, >BD, >55, >62, >5D, >B3, >62	NO ME ALC
DATA	>B0, >74, >D5, >E6, >2A, >12, >27, >32	NO ME ALC
DATA	>BD, >AD, >20, >80	NO ME ALC
DATA	>3E, >4D, >61, >8A, >9D, >5E, >91, >B6	NO ME ALC
DATA	>8D, >A9, >5D, >5B, >66, >D8, >1E, >66	NO ME ALC
DATA	>09, >11, >66, >D4, >F4, >D8, >29, >E6	NO ME ALC
DATA	>8A, >50, >B3, >65, >87, >54, >23, >64	NO ME ALC
DATA	>4B, >86, >ED, >53	NO ME ALC
DATA	>35, >73, >2E, >6D, >86, >77, >99, >2A	NO ME ALC
DATA	>35, >37, >39, >DE, >DA, >88, >45, >92	NO ME ALC
DATA	>E0, >A8, >10, >23, >5E, >B1, >8D, >AD	NO ME ALC
DATA	>6B, >8A, >7B, >2B, >72, >96, >93, >1B	NO ME ALC
DATA	>E6, >9D, >DA, >D9	NO ME ALC
DATA	>41, >AE, >AB, >75, >57, >67, >77, >9A	NO ME ALC
DATA	>6D, >6A, >5B, >8B, >9D, >AD, >8C, >8B	NO ME ALC
DATA	>B7, >19, >6E, >F2, >36, >AA, >D1, >66	NO ME ALC
DATA	>B9, >39, >EA, >9A, >44, >9B, >E6, >D5	NO ME ALC
DATA	>28, >E3, >2A, >5D	NO ME ALC
DATA	>9B, >5F, >93, >AE, >1A, >B7, >5B, >7E	NO ME ALC
DATA	>49, >76, >AE, >D2, >6E, >FB, >39, >DA	NO ME ALC
DATA	>9B, >C8, >8A, >E5, >E7, >6B, >EF, >22	NO ME ALC
DATA	>DD, >56, >90, >83, >7F, >30, >77, >5D	NO ME ALC
DATA	>31, >0A, >FE, >CE	NO ME ALC
DATA	>8A, >75, >A5, >35, >DA, >A7, >72, >8B	NO ME ALC
DATA	>91, >65, >EF, >EF, >A2, >6B, >47, >51	NO ME ALC
DATA	>82, >5F, >80, >AC, >6B, >45, >0D, >36	NO ME ALC
DATA	>29, >DC, >BD, >94, >35, >68, >95, >4A	NO ME ALC
DATA	>F7, >54, >66, >6B	NO ME ALC
DATA	>95, >6A, >6B, >1B, >F0, >A3, >07, >05	NO ME ALC
DATA	>28, >80, >01, >0C, >60, >E0, >01, >00	NO ME ALC
DATA		NO ME ALC
DATA	>95, >6A, >6B, >1B, >F0, >A3, >07, >05	NO ME ALC
DATA	>28, >80, >01, >0C, >60, >E0, >01, >00	NO ME ALC
DATA		NO ME ALC
DATA	>3E, >4D, >61, >8A, >9D, >5E, >91, >B6	NO ME ALC
DATA	>8D, >A9, >5D, >5B, >66, >D8, >1E, >66	NO ME ALC
DATA	>09, >11, >66, >D4, >F4, >D8, >29, >E6	NO ME ALC
DATA	>8A, >50, >B3, >65, >87, >54, >23, >64	NO ME ALC
DATA	>4B, >86, >ED, >53	NO ME ALC

END

PAGE
GROM 7

* THIS GROM RESERVED FOR TITLE CHARACTER DEFINITIONS, TITLE *
* SCREEN FORMAT, SCREEN LOAD DATA, BOTTOM 1/3 PLAYING SCREEN *
* FORMAT. ALL ARE REFERENCED BY GROM 3 *

PRMTB2 DATA #TITCHR
DATA #TITSIT

* TABLE OF POINTERS FOR WALL SCROLL ROUTINE *

GCHRLD DATA #CHRLD
CHRLD DATA #LOAD1
DATA #LOAD2
DATA #LOAD3
DATA #LOAD4
DATA #LOAD5
DATA #LOAD6
DATA #LOAD7
DATA #LOAD8

* THE NEXT 1152 BYTES ARE THE PATTERN DEFINITIONS FOR THE *
* TITLE SCREEN. *

TITCHR DATA >FF,>FF,>FF,>FF,>FF,>FF,>FF,>FE >60
DATA >FF,>FF,>FF,>FF,>F0,>C0,>00,>00
DATA >FF,>F0,>00,>0C,>12,>12,>0C,>00
DATA 0,0,0,0,0,0,0,0 CHARACTER SET 12
DATA >FF,>FF,>FF,>FF,>FF,>FC,>F9,>F0
DATA >F8,>F0,>C0,>B0,0,0,0,0
DATA 0,0,0,0,0,0,0,0
DATA >FF,>FF,>FF,>FF,>FF,>FF,>FF,>FF >67

DATA >F0,>E0,>C0,>B0,>B0,>00,>00,>00 >68
DATA >00,>00,>10,>2B,>2B,>10,>00,>00
DATA >3C,>42,>B1,>B1,>B1,>41,>22,>1C
DATA >FE,>FC,>F8,>F8,>F0,>E0,>C0 CHARACTER SET 13
DATA >38,>44,>B4,>B4,>48,>30,>00,>00
DATA >07,>03,>01,0,0,0,0,0
DATA >FF,>FF,>FF,>FF,>7F,>3E,>1C,>00
DATA >E0,>C0,>B0,0,0,0,0,0 >6F

DATA >FF,>FF,>E0,>C0,>B0,>00,>00,>01 >70
DATA >FF,>FF,0,0,0,0,0,>FB
DATA >FF,>FF,>7F,>3F,>1F,>0F,>0F,>0F
DATA >FF,>FF,>FF,>FF,>FF,>FB,>F0,>E0
DATA >E0,>C0,>C0,>B0,>B0,>FE,>FF,>FF 14
DATA 0,0,0,0,0,0,0,>B0
DATA >00,>00,>00,>60,>90,>90,>60,>00
DATA >00,>04,>06,7,7,7,7,7 >77

DATA 0,0,0,0,>B1,>C3,>E7,>FF >78
DATA >00,>40,>C0,>C0,>C0,>C0,>C0
DATA >03,7,7,7,7,7,7
DATA >FC,>FE,>FE,>FE,>FE,>FE,>FE,>FE 15
DATA >0F,>0F,>0F,>0F,>0F,>0F,>0F,>0F
DATA >C0,>C1,>C3,>C3,>C3,>C3,>C3,>C3

DATA >FF, >07, 3, 3, 3, 3, 3, 3
DATA >C3, >C3, >C3, >C3, >C3, >C3, >C3, >C3 >7F

DATA >C1, >C1, >E1, >E1, >F1, >F1, >B9, >B9 >80
DATA >C0, >C0, >C0, >C0, >C0, >C0, >C0, >C0

DATA 7, 7, 7, 7, 7, 7, 7, 7
DATA >C0, >C0, >C0, >C0, >C0, >C0, >C0, >C0 16

DATA 7, 7, 7, 7, 7, 7, 7, >03
DATA >FE, >FE, >FE, >FE, >FE, >FE, >FE, >FC
DATA >0F, >0F, >0F, >0F, >0F, >0F, >0F, >0F
DATA >C3, >C3, >C3, >C3, >C3, >C1, >C0, >C0 >87

DATA 3, 3, 3, 3, 3, >07, >FF, >FF >88
DATA >C3, >C3, >C3, >C3, >C3, >C3, >C3, >C3

DATA >9D, >9D, >BF, >BF, >87, >87, >83, >83
DATA >C0, >C0, >C0, >C0, >C0, >C0, >C0, >C0

DATA >40, >A0, >40, 0, 0, 0, 0, 0
DATA 0, 0, 0, 0, 0, 0, 0, 0 17

DATA >01, >00, >00, >80, >C0, >E0, >FF, >FF
DATA >FB, 0, 0, 0, 0, 0, >FF, >FF >8F

DATA >0F, >0F, >0F, >1F, >3F, >7F, >FF, >FF >90
DATA >E0, >F0, >FB, >FF, >FF, >FF, >FF, >FF

DATA >FF, >FF, >FE, >80, >80, >C0, >C0, >E0
DATA >80, 0, 0, 0, 0, 0, 0, 0 18

DATA >FB, >E4, >F2, >F2, >F2, >FC, >FB, >FC
DATA >00, >00, >60, >A0, >A0, >60, >00, >00

DATA >70, >88, >84, >84, >48, >30, >00, >00
DATA >FE, >FF, >FF, >FF, >FF, >FF, >FF, >FF >97

DATA >00, >00, >00, >80, >C0, >C0, >E0, >E0 >98
DATA >F0, >FC, >FE, >FF, >FF, >FF, >FF, >FF

DATA 0, 0, 0, 0, >C0, >E0, >F0, >FB
DATA >00, >00, 1, 1, 1, 1, >00, >00

* DATA >7C, >82, >2E, >2A, >2E, >EB, >02, >7C THESE ARE MY INITS IN A CRATER
DATA >7C, >E6, >83, >81, >81, >81, >C3, >7E 19

DATA >FE, >FF, >FF, >FF, >FF, >FF, >FF, >FF
DATA >00, >00, >C0, >F0, >FE, >FF, >FF, >FF >9F

DATA 0, 0, 0, 0, 0, 0, >80, >FE
DATA >0F, >07, >03, 0, 0, 0, 0, 0 >A0

DATA >07, >03, >01, 0, 0, 0, 0, 0
DATA >03, >01, 0, 0, 0, 0, 0, 0

DATA >0F, >0F, >1F, >1F, >3F, >3F, >7F, >FF
DATA >03, >07, >0F, >3F, >FF, >FF, >FF, >FF 20

DATA 0, 0, 0, 0, 0, >01, >03, >03
DATA >03, >03, >01, 0, 0, 0, 0, 0

DATA >FF, >FF, >FF, >7F, >3F, >1F, >07, >01 >A7
DATA 0, 0, 0, 0, 0, >01, >3F, >3F >AB

DATA >FF, >7F, >7F, >7F, >7F, >FF, >FF, >FF
DATA >1F, >0F, >03, 0, 0, 0, 0, 0

DATA >FF, >FF, >FF, >7E, 0, 0, 0, 0
DATA >FF, >FC, >E1, >83, >07, >0F, >0F, >1F

DATA 0, 0, 0, 0, 0, >60, >7F, >7F 21
DATA >00, >00, >01, >07, >0F, >3F, >FF, >FF

DATA >7F, >FF, >FF, >FF, >FF, >FF, >FF, >FF >AF

```

DATA 0, 0, 0, 0, 0, >01, >01, >03          >B0
DATA >7F, >7F, >FF, >FF, >FF, >FF, >FF, >FF
DATA >07, >07, >0F, >1F, >3F, >3F, >7F, >FF
DATA >FF, >FF, >FF, >FF, >FF, >FF, >FF, >FF
DATA >0F, >0F, >0F, >0F, >0F, >0F, >0F, >1F
DATA >FF, >FF, >FF, >7F, >3F, >3F, >1F, >0F
DATA >FF, >FF, >FF, >FF, >7F, >1F, >0F, >07          22
DATA >FF, >FF, >FF, >FF, >EF, >3F, >1F, >0F          >B7

DATA >FF, >3F, >03, 0, 0, 0, 0, 0          >BB
DATA >FF, >FF, >FF, >3F, >0F, >07, >01, >00
DATA >FF, >FF, >FF, >FF, >FF, >FF, >FF, >7F
DATA >5F, >4F, >47, >23, >1C, >00, >00, >00
DATA >FF, >FF, >FF, >FF, >FF, >7F, >3F, >1F
DATA >0F, >0F, >07, >03, >01, >00, >00, >00          23
DATA >FF, >FF, >FF, >FF, >FF, >FF, >FF, >7F
DATA >7F, >3F, >3F, >1F, >1F, >0F, >0F, >07          >BF

DATA 0, 0, 0, 0, 0, >01, >01, >01          >C0
DATA >07, >07, >03, >03, >01, >FF, >FF, >FF
DATA >FF, >FF, >FF, >FF, >FF, >FF, >FF, >01, >01, >01
DATA >FF, >FF, >F0, >F0, >F0, >F0, >F0, >F0, >F0
DATA >FF, >FF, >3F, >3F, >1F, >1F, >1F, >0F          24
DATA >FF, >FF, >E0, >E0, >E0, >E0, >E0, >E0
DATA 0, 0, 0, 0, 0, 0, >03, >03
DATA 0, 0, 0, 0, 0, 0, >FF, >FF          >C7

DATA 0, 0, 0, 0, 0, 0, >FF, >FF          >CB
DATA 3, 3, 3, 3, 3, 3, 3, 3
DATA >C3, >C3, >E7, >E7, >FF, >FF, >DD, >DD
DATA >C1, >C0, >C0, >C0, >C0, >C0, >C0, >C0          25
DATA >FF, 7, 7, 7, 7, 7, 7, 7
DATA >01, >7F, >7F, >7F, >7F, >7F, >7F, >7F
DATA >F0, >F0, >F0, >F0, >F0, >F0, >F0, >F0
DATA >0F, >07, >07, >43, >43, >61, >61, >60          >CF

DATA >E0, >E0, >E0, >E0, >E0, >E0, >E0, >E0          >D0
DATA 3, 3, 3, 3, 3, >00, >00, >00
DATA >FF, >FF, >FF, >FF, >FF, >FF, >01, >01, >01
DATA 3, 3, 3, 3, 3, 3, 3, 3
DATA >99, >99, >81, >81, >81, >81, >81, >81          26
DATA >C0, >C0, >C0, >C0, >C0, >C0, >C0, >C1
DATA 7, 7, 7, 7, 7, 7, 7, >FF
DATA >7F, >7F, >7F, >7F, >7F, >7F, >7F, >01          >D7

DATA >F0, >F0, >F0, >F0, >F0, >F0, >F0, >F0          >DB
DATA >70, >70, >78, >78, >7C, >7C, >7E, >7E
DATA >E0, >60, >60, >20, >20, >00, >00, >00
DATA >00, >00, >00, 3, 3, 3, 3, 3
DATA >01, >01, >01, >FF, >FF, >FF, >FF, >FF          27
DATA >01, >01, >01, 0, 0, 0, 0, 0
DATA >FF, >FF, >FF, >01, >01, >03, >03, >07
DATA >01, >01, >01, >FF, >FF, >FF, >FF, >FF          >DF

DATA >F0, >F0, >F0, >F0, >F0, >F0, >FF, >FF          >EO
DATA >7F, >7F, >7F, >7F, >7F, >7F, >FF, >FF
DATA >00, >00, >80, >80, >C0, >C0, >FF, >FF
DATA >03, >03, 0, 0, 0, 0, 0, 0

```

```

DATA >FF,>FF,0,0,0,0,0,0      28
DATA >FF,>FF,0,0,0,0,0,0
DATA >07,>07,>0F,>0F,>0F,>1F,>1F,>3F
DATA >00,>0F,>18,>21,>23,>23,>27,>1F  >E7

```

```

DATA >7F,>FF,>FF,>FF,>FF,>FF,>FF,>FF  >E8
DATA 0,0,0,0,>01,>03,>07,>1F
DATA >1F,>3F,>7F,>FF,>FF,>FF,>FF,>FF
DATA 0,0,0,0,0,0,0,>3F
DATA >00,>00,>03,>07,>1F,>FF,>FF,>FF      29
DATA >7F,>FF,>FF,>FF,>FF,>FF,>FF,>FF
DATA >38,>64,>84,>84,>84,>44,>3C,>00
* DATA 3,0,0,0,0,0,0,0
DATA >03,>00,>FF,>29,>2F,>AB,>EB,>00      >EF

```

```

****
**      THAT CONCLUDES 1152 BYTES OF TITLE SCREEN CHARACTERS
**      LOAD THOSE IN AT PDT+>300 (START OF CHAR SET 12)
****

```

```

*****
*****
* RINGS NUMBERED OUT TO IN - SET-UP CLOCKWISE *
*****
* RING #1          * COLOR TBL SET #24          * BLK/GRAY *
*****

```

```

LOAD1 DATA >4E,>00,>4E,>40          * RING 1 (1-8), 2 (9-16)
DATA >80,>40,>20,>10,>08,>04,>02,>01 * T/L CNR >C0
DATA >AA,08,>00                    * TOP >C1
DATA >01,>02,>04,>08,>10,>20,>40,>80 * T/R CNR >C2
DATA >AA,08,>00                    * RIGHT >C3
DATA >80,>40,>20,>10,>08,>04,>02,>01 * B/R CNR >C4
DATA >AA,08,>00                    * BOTTOM >C5
DATA >01,>02,>04,>08,>10,>20,>40,>80 * B/L CNR >C6
DATA >AA,08,>00                    * LEFT >C7

```

```

* DATA >AB,>4E,>40,>4E,>00          * RING 2 (1-8), 1 (9-16)
DATA >80,>40,>20,>10,>08,>07,>07,>07 * T/L CNR >C8
DATA >AA,05,>00,>FF,>FF,>FF        * TOP >C9
DATA >01,>02,>04,>08,>10,>E0,>E0,>E0 * T/R CNR >CA
DATA >AA,08,>E0                    * RIGHT >CB
DATA >E0,>E0,>E0,>10,>08,>04,>02,>01 * B/R CNR >CC
DATA >FF,>FF,>FF,>AA,05,>00        * BOTTOM >CD
DATA >07,>07,>07,>08,>10,>20,>40,>80 * B/L CNR >CE
DATA >AA,08,>07                    * LEFT >CF

```

```

* DATA >AB,>4E,>80,>4E,>C0          * RING 3 (1-8), 4 (9-16)
DATA >80,>40,>20,>10,>08,>04,>02,>01 * T/L CNR >D0
DATA >AA,08,>00                    * TOP >D1
DATA >01,>02,>04,>08,>10,>20,>40,>80 * T/R CNR >D2
DATA >AA,08,>00                    * RIGHT >D3
DATA >80,>40,>20,>10,>08,>04,>02,>01 * B/R CNR >D4
DATA >AA,08,>00                    * BOTTOM >D5
DATA >01,>02,>04,>08,>10,>20,>40,>80 * B/L CN. >D6
DATA >AA,08,>00                    * LEFT >D7

```

```

* DATA >AB,>4E,>C0,>4E,>80          * RING 4 (1-8), 3 (9-16)
DATA >80,>40,>20,>10,>08,>07,>07,>07 * T/L CNR >D8
DATA >AA,05,>00,>FF,>FF,>00        * TOP >D9
DATA >01,>02,>04,>08,>10,>E0,>E0,>E0 * T/R CNR >DA

```

```

DATA >AA,08,>60          * RIGHT >DB
DATA >E0,>E0,>E0,>10,>08,>04,>02,>01 * B/R CNR >DC
DATA >00,>FF,>FF,>AA,05,>00    * BOTTOM >DD
DATA >07,>07,>07,>08,>10,>20,>40,>80 * B/L CNR >DE
DATA >AA,08,>06          * LEFT >DF

```

*

```

DATA >AB,>4F,>00,>4F,>40          * RING 5 (1-8), 6 (9-16)
DATA >80,>40,>20,>10,>08,>04,>02,>01 * T/L CNR >E0
DATA >AA,08,>00          * TOP >E1
DATA >01,>02,>04,>08,>10,>20,>40,>80 * T/R CNR >E2
DATA >AA,08,>00          * RIGHT >E3
DATA >80,>40,>20,>10,>08,>04,>02,>01 * B/R CNR >E4
DATA >AA,08,>00          * BOTTOM >E5
DATA >01,>02,>04,>08,>10,>20,>40,>80 * B/L CNR >E6
DATA >AA,08,>00          * LEFT >E7

```

*

```

DATA >AB,>4F,>40,>4F,>00          * RING 6 (1-8), 5 (9-16)
DATA >80,>40,>20,>10,>08,>07,>06,>05 * T/L CNR >EB
DATA >AA,05,>00,>FF,>00,>00    * TOP >E9
DATA >01,>02,>04,>08,>10,>E0,>60,>A0 * T/R CNR >EA
DATA >AA,08,>20          * RIGHT >EB
DATA >A0,>60,>E0,>10,>08,>04,>02,>01 * B/R CNR >EC
DATA >00,>00,>FF,>AA,05,>00    * BOTTOM >ED
DATA >05,>06,>07,>08,>10,>20,>40,>80 * B/L CNR >EE
DATA >AA,08,>04          * LEFT >EF

```

*

```

* TRAC'S
DATA >AB,>4F,>80,>4F,>80          * TRAC'S SAME FOR ALL
DATA >01,>02,>04,>0F,>10,>20,>40,>FF >F0
DATA >FF,>00,>00,>FF,>00,>00,>00,>FF >F1
DATA >FF,>00,>00,>FF,>01,>03,>07,>FF >F2
DATA >FC,>32,>79,>FF,>F9,>F9,>F9,>FE >F3
DATA >3F,>4C,>9E,>FF,>9F,>9F,>9F,>7F >F4
DATA >FF,>00,>00,>FF,>80,>C0,>E0,>FF >F5
DATA >80,>40,>20,>F0,>08,>04,>02,>FF >F6
DATA >AA,08,>00          >F7 OPEN
DATA >AD

```

```

LOAD2 DATA >4E,>40,>4E,>00          * RING 2 (1-8), 1 (9-16)
DATA >80,>40,>20,>10,>0F,>0F,>0F,>0F * T/L CNR >CB
DATA >AA,04,>00,>FF,>FF,>FF,>00    * TOP >C9
DATA >01,>02,>04,>08,>F0,>F0,>F0,>F0 * T/R CNR >CA
DATA >AA,08,>70          * RIGHT >CB
DATA >F0,>F0,>F0,>F0,>08,>04,>02,>01 * B/R CNR >CC
DATA >00,>FF,>FF,>FF,>AA,04,>00    * BOTTOM >CD
DATA >0F,>0F,>0F,>0F,>10,>20,>40,>80 * B/L CNR >CE
DATA >AA,08,>0E          * LEFT >CF

```

*

```

DATA >AB,>4E,>C0,>4E,>80          * RING 4 (1-8), 3 (9-16)
DATA >80,>40,>20,>10,>0F,>0F,>0E,>0D * T/L CNR >DB
DATA >AA,04,>00,>FF,>FF,>00,>00    * TOP >D9
DATA >01,>02,>04,>08,>F0,>F0,>70,>80 * T/R CNR >DA
DATA >AA,08,>30          * RIGHT >DB
DATA >80,>70,>F0,>F0,>08,>04,>02,>01 * B/R CNR >DC
DATA >00,>00,>FF,>FF,>AA,04,>00    * BOTTOM >DD
DATA >0D,>0E,>0F,>0F,>10,>20,>40,>80 * B/L CNR >DE
DATA >AA,08,>0C          * LEFT >DF

```

*

```

DATA >AB,>4F,>40,>4F,>00          * RING 6 (1-8), 5 (9-16)

```

```

DATA >80,>40,>20,>10,>0F,>0C,>0A,>09 * T/L CNR >E8
DATA >AA,04,>00,>FF,>00,>00,>00 * TOP >E9
DATA >01,>02,>04,>08,>F0,>30,>50,>90 * T/R CNR >EA
DATA >AA,08,>10 * RIGHT >EB
DATA >90,>50,>30,>F0,>08,>04,>02,>01 * B/R CNR >EC
DATA >00,>00,>00,>FF,>AA,04,>00 * BOTTOM >ED
DATA >09,>0A,>0C,>0F,>10,>20,>40,>80 * B/L CNR >EE
DATA >AA,08,>08 * LEFT >EF

```

*

```

DATA >AB,>4F,>80,>4F,>80 * TRAC'S SAME FOR ALL
DATA >01,>02,>07,>08,>10,>20,>7F,>80
DATA >FF,>00,>FF,>00,>00,>00,>FF,>00
DATA >FF,>00,>FF,>00,>01,>03,>FF,>0F
DATA >FC,>32,>F9,>F9,>FF,>F9,>F9,>F2
DATA >3F,>4C,>9E,>9F,>FF,>9F,>9F,>4F
DATA >FF,>00,>FF,>00,>80,>C0,>FF,>F0
DATA >80,>40,>E0,>10,>08,>04,>FE,>01
DATA >AA,08,>00
DATA >AD

```

```

LOAD3 DATA >4E,>40,>4E,>00 * RING 2 (1-8), 1 (9-16)
DATA >80,>40,>20,>1F,>1F,>1F,>1E,>1D * T/L CNR >C8
DATA >00,>00,>00,>FF,>FF,>FF,>00,>00 * TOP >C9
DATA >01,>02,>04,>F8,>F8,>F8,>78,>B8 * T/R CNR >CA
DATA >AA,08,>38 * RIGHT >CB
DATA >88,>78,>F8,>F8,>F8,>04,>02,>01 * B/R CNR >CC
DATA >00,>00,>FF,>FF,>FF,>00,>00,>00 * BOTTOM >CD
DATA >1D,>1E,>1F,>1F,>1F,>20,>40,>80 * B/L CNR >CE
DATA >AA,08,>1C * LEFT >CF

```

*

```

DATA >AB,>4E,>C0,>4E,>80 * RING 4 (1-8), 3 (9-16)
DATA >80,>40,>20,>1F,>1F,>1C,>1A,>19 * T/L CNR >DB
DATA >00,>00,>00,>FF,>FF,>00,>00,>00 * TOP >D9
DATA >01,>02,>04,>F8,>F8,>38,>58,>98 * T/R CNR >DA
DATA >AA,08,>18 * RIGHT >DB
DATA >98,>58,>38,>F8,>F8,>04,>02,>01 * B/R CNR >DC
DATA >00,>00,>00,>FF,>FF,>00,>00,>00 * BOTTOM >DD
DATA >19,>1A,>1C,>1F,>1F,>20,>40,>80 * B/L CNR >DE
DATA >AA,08,>18 * LEFT >DF

```

*

```

DATA >AB,>4F,>40,>4F,>00 * RING 6 (1-8), 5 (9-16)
DATA >80,>40,>20,>1F,>18,>14,>12,>11 * T/L CNR >E8
DATA >00,>00,>00,>FF,>AA,04,>00 * TOP >E9
DATA >01,>02,>04,>F8,>18,>28,>48,>88 * T/R CNR >EA
DATA >AA,08,>08 * RIGHT >EB
DATA >88,>48,>28,>18,>F8,>04,>02,>01 * B/R CNR >EC
DATA >AA,04,>00,>FF,>00,>00,>00 * BOTTOM >ED
DATA >11,>12,>14,>18,>1F,>20,>40,>80 * B/L CNR >EE
DATA >AA,08,>10 * LEFT >EF

```

*

```

DATA >AB,>4F,>80,>4F,>80 * TRAC'S SAME FOR ALL
DATA >01,>03,>04,>08,>10,>3F,>40,>80
DATA >FF,>FF,>00,>00,>00,>FF,>00,>00
DATA >FF,>FF,>00,>00,>01,>FF,>07,>0F
DATA >FC,>FE,>79,>F9,>F9,>FF,>F9,>F2
DATA >3F,>7F,>9E,>9F,>9F,>FF,>9F,>4F
DATA >FF,>FF,>00,>00,>80,>FF,>E0,>F0
DATA >80,>C0,>20,>10,>08,>FC,>02,>01

```

DATA >AA, 08, >00

DATA >AD

*****LOAD4 DATA >4E, >40, >4E, >00 * RING 2 (1-8), 1 (9-16)
DATA >80, >40, >3F, >3F, >3F, >3C, >3A, >39 * T/L CNR >CB
DATA >00, >00, >FF, >FF, >FF, >00, >00, >00 * TOP >C9
DATA >01, >02, >FC, >FC, >FC, >3C, >5C, >9C * T/R CNR >CA
DATA >AA, 08, >1C * RIGHT >CB
DATA >9C, >5C, >3C, >FC, >FC, >FC, >02, >01 * B/R CNR >CC
DATA >00, >00, >00, >FF, >FF, >FF, >00, >00 * BOTTOM >CD
DATA >39, >3A, >3C, >3F, >3F, >3F, >40, >80 * B/L CNR >CE
DATA >AA, 08, >38 * LEFT >CF

*

DATA >AB, >4E, >C0, >4E, >80 * RING 4 (1-8), 3 (9-16)
DATA >80, >40, >3F, >3F, >3B, >34, >32, >31 * T/L CNR >DB
DATA >00, >00, >FF, >FF, >AA, 04, >00 * TOP >D9
DATA >01, >02, >FC, >FC, >1C, >2C, >4C, >8C * T/R CNR >DA
DATA >AA, 08, >0C * RIGHT >DB
DATA >8C, >4C, >2C, >1C, >FC, >FC, >02, >01 * B/R CNR >DC
DATA >AA, 04, >00, >FF, >FF, >00, >00 * BOTTOM >DD
DATA >31, >32, >34, >3B, >3F, >3F, >40, >80 * B/L CNR >DE
DATA >AA, 08, >30 * LEFT >DF

*

DATA >AB, >4F, >40, >4F, >00 * RING 6 (1-8), 5 (9-16)
DATA >80, >40, >3F, >30, >2B, >24, >22, >21 * T/L CNR >EB
DATA >00, >00, >FF, >AA, 05, >00 * TOP >E9
DATA >01, >02, >FC, >04, >14, >24, >44, >84 * T/R CNR >EA
DATA >AA, 08, >04 * RIGHT >EB
DATA >84, >44, >24, >14, >04, >FC, >02, >01 * B/R CNR >EC
DATA >AA, 05, >00, >FF, >00, >00 * BOTTOM >ED
DATA >21, >22, >24, >2B, >30, >3F, >40, >80 * B/L CNR >EE
DATA >AA, 08, >20 * LEFT >EF

*

DATA >AB, >4F, >80, >4F, >80 * TRAC'S SAME FOR ALL
DATA >01, >02, >04, >08, >1F, >20, >40, >80
DATA >FF, >00, >00, >00, >FF, >00, >00, >00
DATA >FF, >00, >00, >00, >FF, >03, >07, >0F
DATA >FC, >32, >7F, >F9, >F9, >F9, >FF, >F2
DATA >3F, >4C, >FE, >9F, >9F, >9F, >FF, >4F
DATA >FF, >00, >00, >00, >FF, >C0, >E0, >F0
DATA >80, >40, >20, >10, >FB, >04, >02, >01
DATA >AA, 08, >00
DATA >AD*****
*****LOAD5 DATA >4E, >40, >4E, >00 * RING 2 (1-8), 1 (9-16)
DATA >80, >7F, >7F, >7F, >7B, >74, >72, >71 * T/L CNR >CB
DATA >00, >FF, >FF, >FF, >AA, 04, >00 * TOP >C9
DATA >01, >FE, >FE, >FE, >1E, >2E, >4E, >8E * T/R CNR >CA
DATA >AA, 08, >0E * RIGHT >CB
DATA >8E, >4E, >2E, >1E, >FE, >FE, >FE, >01 * B/R CNR >CC
DATA >AA, 04, >00, >FF, >FF, >FF, >00 * BOTTOM >CD
DATA >71, >72, >74, >7B, >7F, >7F, >7F, >80 * B/L CNR >CE
DATA >AA, 08, >70 * LEFT >CF

*

DATA >AB, >4E, >C0, >4E, >80 * RING 4 (1-8), 3 (9-16)
DATA >80, >7F, >7F, >70, >6B, >64, >62, >61 * T/L CNR >DB

```

DATA >00,>FF,>FF,>AA,05,>00      * TOP      >D9
DATA >01,>FE,>FE,>0E,>16,>26,>46,>86 * T/R CNR >DA
DATA >AA,08,>06                  * RIGHT   >DB
DATA >86,>46,>26,>16,>0E,>FE,>FE,>01 * B/R CNR >DC
DATA >AA,05,>00,>FF,>FF,>00      * BOTTOM   >DD
DATA >61,>62,>64,>68,>70,>7F,>7F,>80 * B/L CNR >DE
DATA >AA,08,>60                  * LEFT    >DF

```

*

```

DATA >AB,>4F,>40,>4F,>00          * RING 6 (1-8), 5 (9-16)
DATA >80,>7F,>60,>50,>48,>44,>42,>41 * T/L CNR >E8
DATA >00,>FF,>AA,06,>00          * TOP     >E9
DATA >01,>FE,>06,>0A,>12,>22,>42,>82 * T/R CNR >EA
DATA >AA,08,>02                  * RIGHT   >EB
DATA >82,>42,>22,>12,>0A,>06,>FE,>01 * B/R CNR >EC
DATA >AA,06,>00,>FF,>00          * BOTTOM   >ED
DATA >41,>42,>44,>48,>50,>60,>7F,>80 * B/L CNR >EE
DATA >AA,08,>40                  * LEFT    >EF

```

*

```

DATA >AB,>4F,>80,>4F,>80          * TRAC'S SAME FOR ALL
DATA >01,>02,>04,>0F,>10,>20,>40,>FF
DATA >FF,>00,>00,>FF,>00,>00,>FF
DATA >FF,>00,>00,>FF,>01,>03,>07,>FF
DATA >FC,>32,>79,>FF,>F9,>F9,>F9,>FE
DATA >3F,>4C,>9E,>FF,>9F,>9F,>9F,>7F
DATA >FF,>00,>00,>FF,>80,>C0,>E0,>FF
DATA >80,>40,>20,>F0,>08,>04,>02,>FF
DATA >AA,08,>00
DATA >AD

```

```

*****
*****

```

```

LOAD6 DATA >4E,>40,>4E,>00          * RING 2 (1-8), 1 (9-16)
DATA >FF,>FF,>FF,>F0,>E8,>E4,>E2,>E1 * T/L CNR >C8
DATA >FF,>FF,>FF,>AA,05,>00          * TOP     >C9
DATA >FF,>FF,>FF,>0F,>17,>27,>47,>87 * T/R CNR >CA
DATA >AA,08,>07                  * RIGHT   >CB
DATA >87,>47,>27,>17,>0F,>FF,>FF,>FF * B/R CNR >CC
DATA >AA,05,>00,>FF,>FF,>FF          * BOTTOM   >CD
DATA >E1,>E2,>E4,>E8,>F0,>FF,>FF,>FF * B/L CNR >CE
DATA >AA,08,>E0                  * LEFT    >CF

```

*

```

DATA >AB,>4E,>C0,>4E,>80          * RING 4 (1-8), 3 (9-16)
DATA >FF,>FF,>E0,>D0,>C8,>C4,>C2,>C1 * T/L CNR >D8
DATA >FF,>FF,>AA,06,>00          * TOP     >D9
DATA >FF,>FF,>07,>0B,>13,>23,>43,>83 * T/R CNR >DA
DATA >AA,08,>03                  * RIGHT   >DB
DATA >83,>43,>23,>13,>0B,>07,>FF,>FF * B/R CNR >DC
DATA >AA,06,>00,>FF,>FF          * BOTTOM   >DD
DATA >C1,>C2,>C4,>C8,>D0,>E0,>FF,>FF * B/L CN. >DE
DATA >AA,08,>C0                  * LEFT    >DF

```

*

```

DATA >AB,>4F,>40,>4F,>00          * RING 6 (1-8), 5 (9-16)
DATA >FF,>C0,>A0,>90,>88,>84,>82,>81 * T/L CNR >E8
DATA >FF,>AA,07,>00            * TOP     >E9
DATA >FF,>03,>05,>09,>11,>21,>41,>81 * T/R CNR >EA
DATA >AA,08,>01                * RIGHT   >EB
DATA >81,>41,>21,>11,>09,>05,>02,>FF * B/R CNR >EC
DATA >AA,07,>00,>FF            * BOTTOM   >ED
DATA >81,>82,>84,>88,>90,>A0,>C0,>FF * B/L CNR >EE

```

DATA >AA, 08, >80 * LEFT >EF

*

DATA >AB, >4F, >80, >4F, >80 * TRAC'S SAME FOR ALL

DATA >01, >02, >07, >08, >10, >20, >7F, >80

DATA >FF, >00, >FF, >00, >00, >00, >FF, >00

DATA >FF, >00, >FF, >00, >01, >03, >FF, >0F

DATA >FC, >32, >F9, >F9, >FF, >F9, >F9, >F2

DATA >3F, >4C, >9E, >9F, >FF, >9F, >9F, >4F

DATA >FF, >00, >FF, >00, >80, >C0, >FF, >F0

DATA >80, >40, >E0, >10, >08, >04, >FE, >01

DATA >AA, 08, >00

DATA >AD

LOAD7 DATA >4E, >00, >4E, >40 * RING 1 (1-8), 2 (9-16)

DATA >80, >40, >20, >10, >08, >04, >02, >01 * T/L CNR >C0

DATA >AA, 07, >00, >FF * TOP >C1

DATA >01, >02, >04, >08, >10, >20, >40, >80 * T/R CNR >C2

DATA >AA, 08, >80 * RIGHT >C3

DATA >80, >40, >20, >10, >08, >04, >02, >01 * B/R CNR >C4

DATA >FF, >AA, 07, >00 * BOTTOM >C5

DATA >01, >02, >04, >08, >10, >20, >40, >80 * B/L CNR >C6

DATA >AA, 08, >01 * LEFT >C7

*

DATA >AB, >4E, >40, >4E, >00 * RING 2 (1-8), 1 (9-16)

DATA >FF, >FF, >E0, >D0, >C8, >C4, >C2, >C1 * T/L CNR >C8

DATA >FF, >FF, >AA, 06, >00 * TOP >C9

DATA >FF, >FF, >07, >0B, >13, >23, >43, >83 * T/R CNR >CA

DATA >AA, >0B, >03 * RIGHT >CB

DATA >83, >43, >23, >13, >0B, >07, >FF, >FF * B/R CNR >CC

DATA >AA, 06, >00, >FF, >FF * BOTTOM >CD

DATA >C1, >C2, >C4, >C8, >D0, >E0, >FF, >FF * B/L CNR >CE

DATA >AA, 08, >C0 * LEFT >CF

*

DATA >AB, >4E, >80, >4E, >C0 * RING 3 (1-8), 4 (9-16)

DATA >80, >40, >20, >10, >08, >04, >02, >01 * T/L CNR >D0

DATA >AA, 07, >00, >FF * TOP >D1

DATA >01, >02, >04, >08, >10, >20, >40, >80 * T/R CNR >D2

DATA >AA, 08, >80 * RIGHT >D3

DATA >80, >40, >20, >10, >08, >04, >02, >01 * B/R CNR >D4

DATA >FF, >AA, 07, >00 * BOTTOM >D5

DATA >01, >02, >04, >08, >10, >20, >40, >80 * B/L CNR >D6

DATA >AA, 08, >01 * LEFT >D7

*

DATA >AB, >4E, >C0, >4E, >80 * RING 4 (1-8), 3 (9-16)

DATA >FF, >C0, >A0, >90, >88, >84, >82, >81 * T/L CNR >D8

DATA >FF, >AA, 07, >00 * TOP >D9

DATA >FF, >03, >05, >09, >11, >21, >41, >81 * T/R CNR >DA

DATA >AA, 08, >01 * RIGHT >DB

DATA >81, >41, >21, >11, >09, >05, >03, >FF * B/R CNR >DC

DATA >AA, 07, >00, >FF * BOTTOM >DD

DATA >81, >82, >84, >88, >90, >A0, >C0, >FF * B/L CNR >DE

DATA >AA, 08, >80 * LEFT >DF

*

DATA >AB, >4F, >00, >4F, >40 * RING 5 (1-8), 6 (9-16)

DATA >80, >40, >20, >10, >08, >04, >02, >01 * T/L CNR >E0

DATA >AA, 07, >00, >FF * TOP >E1

DATA >01, >02, >04, >08, >10, >20, >40, >80 * T/R CNR >E2


```

DATA >AA, 08, >80          * RIGHT    >E3
DATA >80, >40, >20, >10, >08, >04, >02, >01 * B/R CNR >E4
DATA >FF, >AA, 07, >00    * BOTTOM   >E5
DATA >01, >02, >04, >08, >10, >20, >40, >80 * B/L CNR >E6
DATA >AA, 08, >01        * LEFT    >E7

```

*

```

DATA >AB, >4F, >40, >4F, >00          * RING 6 (1-8), 5 (9-16)
DATA >80, >40, >20, >10, >08, >04, >02, >01 * T/L CNR >EB
DATA >AA, 08, >00                    * TOP     >E9
DATA >01, >02, >04, >08, >10, >20, >40, >80 * T/R CNR >EA
DATA >AA, 08, >00                    * RIGHT   >EB
DATA >80, >40, >20, >10, >08, >04, >02, >01 * B/R CNR >EC
DATA >AA, 08, >00                    * BOTTOM   >ED
DATA >01, >02, >04, >08, >10, >20, >40, >80 * B/L CNR >EE
DATA >AA, 08, >00                    * LEFT    >EF

```

*

```

DATA >AB, >4F, >80, >4F, >80          * TRAC'S SAME FOR ALL
DATA >01, >03, >04, >08, >10, >3F, >40, >80
DATA >FF, >FF, >00, >00, >00, >FF, >00, >00
DATA >FF, >FF, >00, >00, >01, >FF, >07, >0F
DATA >FC, >FE, >79, >F9, >F9, >FF, >F9, >F2
DATA >3F, >7F, >9E, >9F, >9F, >FF, >9F, >4F
DATA >FF, >FF, >00, >00, >80, >FF, >E0, >F0
DATA >80, >C0, >20, >10, >08, >FC, >02, >01
DATA >AA, 08, >00
DATA >AD

```

```

*****
*****

```

LOADS

```

DATA >4E, >00, >4E, >40          * RING 1 (1-8), 2 (9-16)
DATA >80, >40, >20, >10, >08, >04, >03, >03 * T/L CNR >C0
DATA >AA, 06, >00, >FF, >FF      * TOP     >C1
DATA >01, >02, >04, >08, >10, >20, >C0, >C0 * T/R CNR >C2
DATA >AA, 08, >C0                * RIGHT   >C3
DATA >C0, >C0, >20, >10, >08, >04, >02, >01 * B/R CNR >C4
DATA >FF, >FF, >AA, 06, >00      * BOTTOM   >C5
DATA >03, >03, >04, >08, >10, >20, >40, >80 * B/L CNR >C6
DATA >AA, 08, >03                * LEFT    >C7

```

*

```

DATA >AB, >4E, >40, >4E, >00          * RING 2 (1-8), 1 (9-16)
DATA >FF, >C0, >A0, >90, >88, >84, >82, >81 * T/L CNR >C8
DATA >FF, >AA, 07, >00          * TOP     >C9
DATA >FF, >03, >05, >09, >11, >21, >41, >81 * T/R CNR >CA
DATA >AA, 08, >01                * RIGHT   >CB
DATA >81, >41, >21, >11, >09, >05, >03, >FF * B/R CNR >CC
DATA >AA, 07, >00, >FF          * BOTTOM   >CD
DATA >81, >82, >84, >88, >90, >A0, >C0, >FF * B/L CNR >CE
DATA >AA, 08, >80                * LEFT    >CF

```

*

```

DATA >AB, >4E, >80, >4E, >C0          * RING 3 (1-8), 4 (9-16)
DATA >80, >40, >20, >10, >08, >04, >03, >03 * T/L CNR >D0
DATA >AA, 06, >00, >FF, >FF      * TOP     >D1
DATA >01, >02, >04, >08, >10, >20, >C0, >C0 * T/R CNR >D2
DATA >AA, 08, >C0                * RIGHT   >D3
DATA >C0, >C0, >20, >10, >08, >04, >02, >01 * B/R CNR >D4
DATA >FF, >FF, >AA, 06, >00      * BOTTOM   >D5
DATA >03, >03, >04, >08, >10, >20, >40, >80 * B/L CNR >D6
DATA >AA, 08, >03                * LEFT    >D7

```

*

```

DATA >AB,>4E,>C0,>4E,>80      * RING 4 (1-8), 3 (9-16)
DATA >80,>40,>20,>10,>08,>04,>02,>01 * T/L CNR >DB
DATA >AA,08,>00              * TOP >D9
DATA >01,>02,>04,>08,>10,>20,>40,>80 * T/R CNR >DA
DATA >AA,08,>00              * RIGHT >DB
DATA >80,>40,>20,>10,>08,>04,>02,>01 * B/R CNR >DC
DATA >AA,08,>00              * BOTTOM >DD
DATA >01,>02,>04,>08,>10,>20,>40,>80 * B/L CNR >DE
DATA >AA,08,>00              * LEFT >DF

```

*

```

DATA >AB,>4F,>00,>4F,>40      * RING 5 (1-8), 6 (9-16)
DATA >80,>40,>20,>10,>08,>04,>03,>03 * T/L CNR >E0
DATA >AA,06,>00,>FF,>00      * TOP >E1
DATA >01,>02,>04,>08,>10,>20,>C0,>C0 * T/R CNR >E2
DATA >AA,08,>40              * RIGHT >E3
DATA >C0,>C0,>20,>10,>08,>04,>02,>01 * B/R CNR >E4
DATA >00,>FF,>AA,06,>00      * BOTTOM >E5
DATA >03,>03,>04,>08,>10,>20,>40,>80 * B/L CNR >E6
DATA >AA,08,>02              * LEFT >E7

```

*

```

DATA >AB,>4F,>40,>4F,>00      * RING 6 (1-8), 5 (9-16)
DATA >80,>40,>20,>10,>08,>04,>02,>01 * T/L CNR >E8
DATA >AA,08,>00              * TOP >E9
DATA >01,>02,>04,>08,>10,>20,>40,>80 * T/R CNR >EA
DATA >AA,08,>00              * RIGHT >EB
DATA >80,>40,>20,>10,>08,>04,>02,>01 * B/R CNR >EC
DATA >AA,08,>00              * BOTTOM >ED
DATA >01,>02,>04,>08,>10,>20,>40,>80 * B/L CNR >EE
DATA >AA,08,>00              * LEFT >EF

```

*

```

DATA >AB,>4F,>80,>4F,>80      * TRAC'S SAME FOR ALL
DATA >01,>02,>04,>08,>1F,>20,>40,>80
DATA >FF,>00,>00,>00,>FF,>00,>00,>00
DATA >FF,>00,>00,>00,>FF,>03,>07,>0F
DATA >FC,>32,>7F,>F9,>F9,>F9,>FF,>F2
DATA >3F,>4C,>FE,>9F,>9F,>9F,>FF,>4F
DATA >FF,>00,>00,>00,>FF,>C0,>E0,>F0
DATA >80,>40,>20,>10,>F8,>04,>02,>01
DATA >AA,08,>00
DATA >AD

```

```

*****
* TITLE SCREEN OUTPUT CHARACTERS (FORMAT OF SCREEN) *
*****

```

```

TITSIT DATA >67,>67,>67,>67,>67,>67,>67,>67 * LINE 4
DATA >67,>67,>67,>67,>60,>61,>62,>A0
DATA >B7,>B8,>B9,>BA,>67,>67,>67,>67
DATA >67,>67,>67,>67,>67,>67,>67,>67

```

*

```

DATA >67,>67,>67,>67,>67,>67,>67,>67 * LINE 5
DATA >67,>67,>67,>64,>65,>66,>66,>66
DATA >A1,>B6,>B3,>BB,>BC,>67,>67,>67
DATA >67,>67,>67,>67,>67,>67,>67,>67

```

*

```

DATA >67,>67,>67,>67,>67,>67,>67,>67 * LINE 6
DATA >67,>67,>60,>68,>66,>69,>66,>66
DATA >6A,>A2,>B5,>B3,>BD,>BE,>67,>67
DATA >67,>67,>67,>67,>67,>67,>67,>67

```

*

```

DATA >67,>67,>67,>67,>67,>67,>67,>67 * LINE 7
DATA >67,>67,>6B,>6C,>66,>66,>66,>66
DATA >66,>66,>B4,>EE,>B3,>BF,>67,>67
DATA >67,>67,>67,>67,>67,>67,>67,>67

```

*

```

DATA >67,>67,>6D,>6E,>6F,>67,>70,>71 * LINE 8
DATA >72,>73,>74,>75,>66,>66,>76,>66
DATA >66,>66,>A3,>B3,>C0,>C1,>C2,>C3
DATA >C4,>C5,>67,>C6,>C7,>C8,>67,>67

```

*

```

DATA >67,>67,>77,>78,>79,>67,>7A,>7B * LINE 9
DATA >7C,>7D,>7E,>7F,>80,>81,>66,>66
DATA >A5,>A4,>C9,>CA,>CB,>CC,>CD,>CE
DATA >CF,>D0,>67,>D1,>D2,>67,>67,>67

```

*

```

DATA >67,>67,>82,>67,>83,>67,>84,>85 * LINE 10
DATA >86,>87,>88,>89,>8A,>8B,>66,>8C
DATA >A6,>A7,>D3,>D4,>D5,>D6,>D7,>D8
DATA >D9,>DA,>67,>DB,>DC,>67,>67,>67

```

*

```

DATA >67,>67,>82,>67,>83,>67,>8E,>8F *LINE 11
DATA >90,>91,>92,>93,>66,>66,>66,>66
DATA >66,>A8,>A9,>B3,>DD,>DE,>DF,>E0
DATA >E1,>E2,>67,>E3,>E4,>E5,>67,>67

```

*

```

DATA >67,>67,>67,>67,>67,>67,>67,>67 * LINE 12
DATA >67,>67,>94,>66,>66,>95,>66,>96
DATA >66,>AA,>AB,>AC,>B3,>E6,>67,>67
DATA >67,>67,>67,>67,>67,>67,>67,>67

```

*

```

DATA >67,>67,>67,>67,>67,>67,>67,>67 * LINE 13
DATA >67,>67,>97,>98,>66,>66,>66,>66
DATA >66,>AD,>AE,>AF,>E7,>E8,>67,>67
DATA >67,>67,>67,>67,>67,>67,>67,>67

```

*

```

DATA >67,>67,>67,>67,>67,>67,>67,>67 * LINE 14
DATA >67,>67,>67,>99,>9A,>9B,>9C,>66
DATA >B0,>B1,>B3,>E9,>EA,>67,>67,>67
DATA >67,>67,>67,>67,>67,>67,>67,>67

```

*

```

DATA >67,>67,>67,>67,>67,>67,>67,>67 * LINE 15
DATA >67,>67,>67,>67,>9D,>9E,>9F,>66
DATA >B2,>EB,>EC,>ED,>67,>67,>67,>67
DATA >67,>67,>67,>67,>67,>67,>67,>67

```

*

```

** THAT WAS 384 BYTES OF SIT CHARS FOR THE TITLE SCREEN
** LOAD THOSE AT SIT >80 AFTER AND ALL >20 GPL COMMAND

```

*

END