```
.TITLE ALWELG-ALIENS WELL GAME MAINLINE
        .SBTTL ***************************
        .SBTTL *
        .SBTTL *PROGRAMMER  DFT
        .SBTTL *MODULE  ALWELG
        .SBTTL *FUNCTION  PERFORMS ALIENS WELL GAME FUNCTION
        .SBTTL *
        .SBTTL ***************************
        .INCLUDE ALCOMN
        .PAGE
        .GLOBL INEWAV,INEWLI,MOVCUR,INIDSP,NEWAV2,UPSCOR,GETCUR,INIRA0
        .GLOBL SWAPEN,PLAY
        .GLOBL SLAUNC,EXSNON,ESLSON,SBOING,PRSTAR,SOUTS3,PRBOOM,INBOOM
        .GLOBL INIRAT,PRORAT,LEVEL,CCEXPL,CIEXPL,CPEXPL,IPEXPL
        .GLOBL SELICO,PLDROP,BONSCO,SOUTS2,SOUTS3,INICOL,S3SWAR
        .GLOBL PPSPXI,CPSPXI,FPSPXI,PULSTR,D70MSK,PULSTO
        .GLOBL QCHKS2,QCHKS3,QCHKS4,QCHKS5
        .ASECT
        . 9000
        .BYTE 02,0BB,5A,30          MORSE CODE ATARI
        .BYTE 50,0EE,3D,0A8
CHKSM2          .BYTE QCHKS2
        .SBTTL INITIALIZE - MAINLINE
INEWAV                              NEW WAVE
        JSR CONTOUR
        JSR INIENE                  INITIALIZE NYMPHS, ENEMY LINES
        JSR INIOBJ                  INITIALIZE OBJECTS
        JSR INISUZ                  NEW SUPERZAPPER
        LDA I,0FA
        STA EYH
        LDA I,0                     CURSOR STARTS AT TOP, NOT DESCENDING
        STA CURMOD
        STA EYL
        LDA I,CDPLAY
        STA QDSTATE
        RTS
INEWLI                              NEW LIFE
        JSR INICUR                  INITIALIZE CURSOR
        JSR CONTOUR                 SET SKILL LEVEL ACC TO WAVE
INIOBJ  JSR INICHA                  DEACTIVATE CHARGES
        JSR INIINV                  DEACTIVATE INVADERS
        JSR ININYM                  INITIALIZE NYMPHS
        JSR INIEXP                  DEACTIVATE EXPLOSIONS
        JSR CLRPOT                  CLEAR POT
        JSR INIDSP                  INITIALIZE DISPLAY
        LDA I,-1
        STA BOFLASH                 BONUS FLASHER CLEARED
        STA PULSON
        LDA I,0                     CLEAR ENEMY SPIKE COUNTER
        STA ELICNT
        RTS
        .PAGE

        .SBTTL  INITIALIZE-NEW WAVE PART 2

NEWAV2
        LDA I,ILINLIY
        STA CURSY
        LDA I,0
        STA TEMPO
```

```
        STA TEMP2
        LDA ZADEST
        STA TEMP1
        IFMI
        DEC TEMP2
        ENDIF
        LDX I,1
        BEGIN                   CALCULATE Z INCREMENT
        LDA TEMP1
        ASL
        ROR TEMP1
        ROR TEMP0
        DEX
        MIEND
        LDA TEMP0               UPDATE Z CENTER
        CLC
        ADC ZADEST+1
        STA ZADEST+1
        LDA TEMP1
        ADC ZADJL
        STA ZADJL
        LDA TEMP2
        ADC ZADJL+1
        STA ZADJL+1
        LDA EYL                 MOVE EYE CLOSER TO WELL
        CLC
        ADC I,18
        STA EYL
        LDA EYH
        ADC I,0
        STA EYH
        CMP I,0FC
        IFCS
        LDA I,1                 TURN OFF STAR FIELD
        STA PLAGRO
        ENDIF
        LDA EYL                 CALCULATE EYE-DESTINATION DELTA
        SEC
        SBC EYLDES
        LDA EYH
        IFNE
        SBC I,0FF
        ENDIF
        IFEQ                    PAST DESTINATION
        LDA EYLDES              YES STOP AT DEST
        STA EYL
        LDA I,0FF
        STA EYH
        LDA I,CPLAY             GO PLAY GAME
        BIT QSTATUS
        IFPL                    ATTRACT
        LDA I,CENDGA            YES. END IT
        ENDIF
        STA QSTATE
        LDX PLAYUP
        LDA I,0
        STA X,BONUS             CLEAR BONUS
        ENDIF
        LDA I,0FF               REQUEST WELL PIC UPDATE
        STA ROTDIS
```

```
        JMP MOVCUR                  UPDATE CURSOR POSITION
        .PAGE
        .SBTTL  INITIALIZE-PREPARE FOR SKILL LEVEL REQUEST STATE
INIRA0
        LDA HIWAVE                  YES. SET START LEVEL ODD HIGHEST LEVEL
                                    ACC HIGHEST LEVEL  -1  COMPLETED IN LAST GAME
        LDX I,LEVELE-LEVEL
        BEGIN                       LOOP FROM HIGHEST CHOICE TO LOWEST
        DEX
        CMP X,LEVEL
        CSEND                       EXIT WHEN WAVE IN TABLE   HIGHEST LEVEL LAST GAME
        LDY I,4
        LDA OPTIN3
        AND I,4
        IFNE                        MAX MIN TIED TO HI SCORE OPTION
        LDA HSCORH+21.              YES. GET MSB OF HIGH SCORE
        CMP I,30
        IFCS                         300000
        INY                         YES.
        ENDIF
        CMP I,50
        IFCS                         500000
        INY                         YES.
        ENDIF
        CMP I,70
        IFCS                         700000
        INY                         YES.
        ENDIF
        ENDIF
        LDA OPTIN1
        AND I,43
        CMP I,40
        IFEQ                        SALES MODE
        LDY I,1B                    YES. ANYTHING GOES
        ENDIF
        STY TEMPO                   NEW MAX MIN
        CPX TEMPO
        IFCC                        PLAYER HI LEVEL   MAX MIN
        LDX TEMPO                   YES. USE MAX MIN FOR RIGHT LIMIT
        ENDIF
        STX HIRATE                  MAX INDEX INTO LEVEL TABLE
        LDA QSTATUS
        IFMI                        ATTRACT
        LDA I,0                     NO
        STA HIWAVE
        ENDIF
INIRAT
        LDX NEWPLA
        STX PLAYUP                  YES
        IFNE                        SPECIAL CASE FOR 2ND PLAYER
        JSR SWAPEN                  SWAP 1ST PLAYER S ENEMIES OUT
        ENDIF
        LDA I,4                     SET UP DEFAULT LEVELS  LEFT   RIGHT SIDES
        STA RITSID
        LDA I,0FF                   STOP RUMBLE
        STA EYH
        LDA I,0                     INITIALIZE CURSOR
        STA CURSL1
        STA CURSPO
        STA LEFSID
```

```
        STA TIMHIS                  NO ATTRACT DELAY
        LDX QSTATUS
        IFMI                        ATTRACT
        LDA I,SECOND                NO
        STA TIMHIS
        LDA I,OFF
        STA WELTYP                  PREVENT WRAP
        LDA I,CREQRAT               GO TO REQUEST
        STA QSTATE                  RATE STATE
        LDA I,CDREQRA
        STA QDSTATE                 REQUEST RATE DISPLAY STATE
        LDA I,0
        STA CURWAV                  TO GET 1ST COLORS
        JSR INICOL
        LDA I,10                    START TIMER
        ENDIF
        STA QTMPAUS
        JSR CLRPOT                  CLEAR POT
                                    FALL INTO PRORAT STATE
        .SBTTL  INITIALIZE-SET SKILL LEVEL

PRORAT                              UPDATE TIMER
        DEC TIMHIS
        IFMI                        ANOTHER SECOND DONE
        SED                         YES
        LDA QTMPAUS                 DECREMENT # SECONDS
        SEC
        SBC I,1
        STA QTMPAUS
        CLD
        IFMI                        SECONDS LEFT AT 0
        LDA I,MFIRE                 YES. AUTO CHOOSE
        STA SWFINA
        ENDIF
        CMP I,3
        IFEQ
        JSR S3SWAR                  3 SECONDS WARNING
        ENDIF
        LDA I,SECOND                RESTART FRACTIONAL SECONDS TIMER
        STA TIMHIS
        ENDIF
        JSR GETCUR                  UPDATE CURSOR POSITION
        LDA I,MSUZA MFIRE
        LDY QTMPAUS
        CPY I,8
        IFCC
        LDA I,MSUZA MFIRE MSTRT1 MSTRT2
        ENDIF
        AND SWFINA
        IFNE                        PLAYER SELECTING THIS LEVEL
        LDA I,0
        STA SWFINA
        LDA CURSL1                  YES. USE LEVEL FOR THIS PLAYER
        TAY
        LDX PLAYUP
        STA X,BONUS
        LDA Y,LEVEL
        BIT QSTATUS
        IFPL                        ATTRACT
        LDY I,1
```

```
        STY LIVES1
        LDA RANDOM                  YES. CHOOSE FROM 1ST 8 LEVELS
        AND I,7
        ENDIF
        STA X,WAVEN1
        STA CURWAV
        JSR INICOL
        JSR CONTOUR
        JSR INIENE                  INITIALIZE ENEMY
        JSR INISUZ                  NEW SUPERZAPPER
        LDA I,CNEWLIF               GO ON TO GAME PLAY
        STA QSTATE
        JSR CLRPOT                  CLEAR POT
        ENDIF
        LDA SWFINA
        AND I, C MFAKE MFIRE MSUZA MSTRT1 MSTRT2
        STA SWFINA                  CLEAR  SWITCHES NOT PROCESSED  FLAG
        RTS

        .SBTTL  BONUS SCORE DETERMINATION
 INPUT  ACC BONUS LEVEL INDEX
 OUTPUT TEMP0,1, 2 BONUS POINTS
        ACC,X DESTROYED

BONSCO  ASL
        TAX
        LDA I,0                     LSB ALWAYS 0
        STA TEMP0
        LDA X,BONPTM
        STA TEMP1
        LDA X,BONPTH
        STA TEMP2
        RTS
BONPTM  .WORD 0,60,160,320,540,740,940,1140,1340
        .WORD 1520,1700,1880,2080,2260,2480,2660,3000,3400
        .WORD 3820,4150,4390,4720,5310,5810
        .WORD 6240,6560,7660,8980
BONPTH   BONPTM+1
LEVEL    TABLE OF LEVEL #S -1  FOR RATING DISPLAY
        .BYTE 0,2,4,6,8,0A,0C,0E,10,13,21.,17,25.,1B,30.,32.,23,27,2B,46.,48.,51.,5
        .BYTE 72.,80.
LEVELE
        .BYTE 0FF                           END OF TABLE FLAG
        .PAGE
        .SBTTL INITIALIZE - CURSOR
INICUR
        LDA I,0E                    INITIALIZE CURSOR POSITION
        STA CURSL1
        LDA I,0F0
        STA CURSP0
        LDA I,0
        STA CURMOD
        LDA I,0F
        STA CURSL2
        LDA I,ILINLIY
        STA CURSY
        RTS
        .SBTTL INITIALIZE - NYMPHS

 INITIALIZE NYMPHS
```

```
INIENE  LDA NWNYMC                          INITIALIZE FOR NEW WAVE  NYMPH COUNT + ENE
        STA NYMCOU
        LDA NWTELI                  INITIALIZE ENEMY LINES HIGHT
        .SBTTL  INIT ENEMY LINES
                                    ACC INITIAL HEIGHT
        LDX I,NLINES-1
        BEGIN
        STA X,LINEY
        DEX
        MIEND
        RTS

ININYM
        LDA I,0
        LDX I,NNYMPH-1
        BEGIN
        STA X,NYMPY
        DEX
        MIEND
        LDX NYMCOU
        DEX
        BEGIN                       ON 8 Y LEVELS
        LDA RANDOM
        AND I,0F
        STA X,NYMPL
        TXA
        ASL
        ASL
        ASL
        ASL
        ORA X,NYMPL
        IFEQ
        LDA I,0F
        ENDIF
        STA X,NYMPY
        DEX
        MIEND
        RTS
        .SBTTL INITIALIZE - INVADERS

 INITIALIZE INVADERS

INIINV  LDX I,NINVAD-1
        LDA I,0
        BEGIN                       LOOP FOR EACH INVADER
        STA X,INVAY                 DEACTIVATE
        DEX
        MIEND
        STA INMCOU
        STA INCCOU
        STA SPINCO
        STA FLIPCO
        STA TANKCO
        STA PULSCO
        STA FUSECO
        RTS
        .SBTTL INITIALIZE - CHARGES
INICHA
        LDA I,0
```

```
        LDX I,NCHARG-1
        BEGIN                       LOOP FOR EACH CHARGE
        STA X,CHARY                 DEACTIVATE CHARGE
        DEX
        MIEND
        STA CHACOU
        STA ESHCOU
        RTS
        .SBTTL  INITIALIZE EXPLOSIONS
INIEXP
        LDX I,NEXPLO-1
        LDA I,0
        BEGIN
        STA X,EXPLOY
        DEX
        MIEND
        STA EXPCOU
        RTS
                                    CLEARS POTS

CLRPOT  LDA I,0
        STA TBHD
        RTS
SWAPEN
        LDX I,SAVEND-SAVEP-1
        BEGIN                       LOOP FOR EACH BYTE OF PLAYER S SPECIAL PARAMETERS
        LDA X,ACTIP                 SWAP ACTIVE TO SAVE AREAS
        LDY X,SAVEP
        STA X,SAVEP
        TYA
        STA X,ACTIP
        DEX
        MIEND
        RTS
        .PAGE
        .SBTTL  INITIALIZE-SET SKILL LEVEL FOR WAVE
CONTOUR

 PARAMETER TABLES DATA STRUCTURE

        BYTE 1 START WAVE
        BYTE 2 END WAVE
        BYTE 0 TYPE OF ENCODING
        BYTE 3 PARAMETERS

 TYPES
T1 2    1 BYTE IN PARAMETER FIELD GOES FOR ALL WAVES IN RANGE
TZ 4    1 BYTE IN PARAMETER FIELD FOR EACH WAVE IN RANGE
TE 0    EOT  RETURN WITH 0
TZANDF 6  AND CURRENT WAVE WITH F, THE DO TZ
TA 8  AND BYTE 4 TO BYTE 3 FOR EACH LEVEL
TB 0A  ADD BYTE 3 TO WINVIN
TR 0C ALTERNATE BETWEEN BYTES 3   4
        LDA CURWAV
        CMP I,98.
        IFCS
        LDA RAND02
        AND I,1F
        ORA I,40
        ENDIF
```

```
        STA TEMP2
        INC TEMP2
        LDX I,WTABEND-WTABLE-1
        STX INDEX1
        BEGIN                           LOOP FOR EACH TABLE ENTRY
        LDX INDEX1
        LDA X,WTABLE
        STA INDYHI
        LDA X,WTABLE-1
        STA INDYLO                      SET UP POINTER TO BYTE TO BE SET UP
        LDA X,WTABLE-2
        STA TEMP4
        LDA X,WTABLE-3
        STA TEMP3                       SET UP POINTER TO ARRAY OP PARAMETERS
        LDA I,1
        STA INDEX2                      SET UP START RANGE COUNTER
        LDY I,0                         SET UP TABLE POINTER
        BEGIN                           LOOP UNTIL CURRENT WAVE IS FOUND
        LDA NY,TEMP3
        STA TYPCOD                      GET TYPE OF RECORD
        BEQ TEXIT                       EXIT ON EOT TYPE CODE WITH 0
        LDA TEMP2
        INY
        CMP NY,TEMP3
        INY
        IFCS                            IS CURRENT WAVE  START WAVE OF RANGE
        CMP NY,TEMP3                    YES.
        IFEQ                              END WAVE OF RANGE
        CLC
        ENDIF
        IFCC
        INY
        JSR DOTYPE                      YES. GET PARAMETER FROM RECORD
        JMP TEXIT                       EXIT LOOP
        ENDIF
        ENDIF
        JSR DONEXT                      DO. UP POINTER TO NEXT RECORD
        CLC
        CSEND                           ALWAYS LOOP
TEXIT
        LDY I,0                         GOT PARAMETER
        STA NY,INDYLO                   SAVE IT
        LDA INDEX1
        SEC
        SBC I,4
        STA INDEX1                      UPDATE MASTER TABLE POINTER
        CMP I,0FF
        EQEND

        .SBTTL  EASY - MED - HARD OPTIONS

ZEASY    1
ZHARD    2
        LDA OPTIN3
        AND I,3
        CMP I,ZEASY
        IFEQ                        EASY
        DEC WCHAMX                  YES. LESS ENEMY SHOTS
        LDA WINVIL
        EOR I,0FF
```

```
        LSR
        LSR
        LSR
        ADC WINVIL
        STA WINVIL              DECREASE SPEEDS BY 1/8
        LDA CURWAV
        CMP I,17.
        IFCC
        DEC WTTFRA              DECREASE FLIP RATE AT TOP
        ENDIF
        ELSE
        CMP I,ZHARD
        IFEQ                    HARD
        INC WCHAMX              YES. MORE ENEMY SHOTS UP TO 4
        LDA WCHAMX
        CMP I,3
        IFCS
        LDA I,3
        STA WCHAMX
        ENDIF
        LDA WINVIL              INCREASE SPEED BY 1/8
        LSR
        LSR
        LSR
        ORA I,0E0
        ADC WINVIL
        STA WINVIL
        LDA NWNYMC              INCREASE ATTACK BY 1/8
        LSR
        LSR
        LSR
        ADC NWNYMC
        STA NWNYMC
        LDA WPULFI
        ORA I,ZFIRYE
        STA WPULFI              PULSARS FIRE
        ENDIF
        ENDIF
        LDA WINVIL+ZABTRA       SPINNER
        JSR TIMES8
        STA WINVIL+ZABTRA       SPEED  FRAC
        STY WINVIN+ZABTRA       SPEED  INT
        STX ENSIZE+ZABTRA       COLLISION RANGE
        LDA WCHARL
        JSR TIMES8              ENEMY SHOT
        STA WCHARL
        STY WCHARIN
        STX CHACHA              CHARGE CHARGE COLLISION RANGE
        LDA WINVIL
        JSR TIMES8
        STA WINVIL
        STA WINVIL+ZABTAN
        STY WINVIN+ZABTAN
        STY WINVIN
        STX ENSIZE+ZABFLI       CHARGE INVADER COLLISION RANGE
        STX ENSIZE+ZABTAN
        STX ENSIZE+ZABPUL
        LDA WINVIL
        ASL
        STA WFUSIL
```

```
        LDA WINVIN
        ROL
        STA WFUSIH                  FUSE INC 2X INVADER SPEED
        LDA I, PCVELO+3 /2
        STA ENSIZE+ZABFUS
        LDA I,0A0
        STA WINVIL+ZABPUL
        LDA I,0FE
        STA WINVIN+ZABPUL
        LDA I,ZCARFL
        STA WTACAR+1
        STA WTACAR+0
        RTS
 INPUT  ACC SPEED  SIGNED
 OUTPUT          ACC LOW BYTE OF SPEED
                 Y HI BYTE OF SPEED  SIGN EXT
                 X COLLISION RANGE WITH PC
                 TEMP0 TRASHED
TIMES8
        LDY I,0FF                   ALL SPEEDS ARE MINUS SO START SIGN
        STY TEMP0                   EXTEND AT ALL-
        ASL
        ROL TEMP0
        ASL
        ROL TEMP0
        ASL
        ROL TEMP0                   X 8
        LDY TEMP0
        PHA                         SAVE RESULT
        TYA                         COLLISION RANGE AVERAGE OF
        EOR I,0FF                   ABS VAL OF SPEEDS.
        CLC
        ADC I,PCVELO+1+1+2
        LSR
        TAX
        PLA
        RTS
        .PAGE
        .SBTTL  SKILL CONTOUR TABLES
                                    FRAMES UNTIL INVADER CAN FIRE  28 PER SECOND
TCHARFR
        .BYTE TA,1,20.,80.,-3
        .BYTE T1,21.,64.,20.
        .BYTE T1,65.,99.,10.
TCHAMX  .BYTE TZ,1,9,1,1,1,2,3,2,2,3,3    ADD 1
        .BYTE T1,10.,64.,2
        .BYTE T1,65.,99.,3

                                    ENEMY SHOT INCREMENT
TINVIN
        .BYTE TA,1,8,-44.,-5
        .BYTE TZ,9,16.,-81.,-84.,-84.,-84.,-88.,-92.,-96.,-96.
        .BYTE TA,17.,25.,-81.,-3
        .BYTE TA,26.,32.,-99.,-3
        .BYTE TA,33.,39.,-108.,-3
        .BYTE TA,40.,48.,-110.,-1
        .BYTE TA,49.,64.,-120.,-1
        .BYTE TR,65.,99.,-160.,-191.
TCHARIN
        .BYTE TB,1,99.,-64.
```

```
TSPIIN  .BYTE TB,1,20.,0
        .BYTE TB,21.,32.,-48.
        .BYTE TB,33.,48.,-40.
        .BYTE TB,49.,99.,-48.
WPULPOT                         PULSAR POTENCY HEIGHT
        .BYTE T1,1,32.,0A0
        .BYTE T1,33.,64.,0A0
        .BYTE T1,65.,99.,0C0
WPULTIM                         PULSAR TIMER INCREMENT
        .BYTE T1,1,48.,4
        .BYTE T1,49.,64.,6
        .BYTE T1,65.,99.,8
WWTAC2
        .BYTE T1,1,32.,ZCARFL
        .BYTE T1,33.,40.,ZCARFU
        .BYTE T1,41.,99.,ZCARPU
WWTAC3
        .BYTE T1,1,48.,ZCARFL
        .BYTE T1,49.,99.,ZCARFU
WSPIMI  .BYTE TZ,1,4,0,0,0,1
        .BYTE T1,5,16.,2
        .BYTE T1,17.,19.,0
        .BYTE T1,20.,32.,1
        .BYTE T1,35.,39.,1
        .BYTE T1,44.,99.,1
        .BYTE TE
WSPIMX  .BYTE TZ,1,6,0,0,0,2,3,4
        .BYTE T1,7,10.,4
        .BYTE T1,11.,16.,3
        .BYTE T1,20.,25.,2
        .BYTE TZ,26.,32.,1,2,2,2,1,1,2
        .BYTE T1,35,39.,1
        .BYTE T1,43.,99.,1
        .BYTE TE
WFLIMI  .BYTE T1,1,4,1
        .BYTE T1,5,99.,0
        .BYTE TE
WFLIMX  .BYTE T1,1,4,4
        .BYTE T1,5,16.,5
        .BYTE T1,17.,19.,3
        .BYTE T1,20.,25.,4
        .BYTE T1,26.,99.,5
        .BYTE TE
WTANMI  .BYTE TZ,1,4,0,0,1,0
        .BYTE T1,5,16.,1
        .BYTE T1,17.,32.,1
        .BYTE T1,33.,39.,1
        .BYTE T1,40.,99.,1
        .BYTE TE
WTANMX  .BYTE TZ,1,5,0,0,1,0,1
        .BYTE T1,6,16.,2
        .BYTE T1,17.,26.,1
        .BYTE T1,27.,32.,1
        .BYTE T1,33.,44.,2
        .BYTE T1,45.,99.,3
        .BYTE TE
WPULMI
        .BYTE T1,17.,32.,2
        .BYTE T1,33.,99.,1
        .BYTE TE
```

```
WPULMX
        .BYTE TZ,17.,32.,5,3,2,2,2,2,2,2,2,2,2,2,2,3,4,2
        .BYTE T1,33.,99.,3
        .BYTE TE
WFUSMI
        .BYTE T1,11.,16.,1
        .BYTE T1,22.,25.,1
        .BYTE T1,27.,99.,1
        .BYTE TE
WFUSMX
        .BYTE T1,11.,16.,1
        .BYTE T1,22.,25.,1
        .BYTE T1,27.,32.,1
        .BYTE T1,33.,39.,4
        .BYTE T1,40.,99.,3
        .BYTE TE
PN 40.
PC 20.
TPUCHDE
        .BYTE TZ,17.,18.,PN,PC
        .BYTE TR,19.,32.,PC,PN
        .BYTE TA,33.,39.,20.,-1
        .BYTE TR,40.,99.,20.,10.
        .BYTE TE
TWFUSC
        .BYTE TR,17.,32.,0,40
        .BYTE TR,33.,48.,40,0C0
        .BYTE T1,49.,99.,0C0
        .BYTE TE
TFUFRQ  .BYTE T1,1,16.,220.
        .BYTE T1,17.,39.,192.
        .BYTE TA,40.,64.,192.,1
        .BYTE T1,65.,99.,230.
TINVMX  .BYTE T1,1,99.,6
TELIHI  .BYTE TZANDF,1,99.,0,0,0,0E0,0D8,0D4,0D0,0C8,0C0,0B8,0B0,0A8,0A0,0A0,0A0,0A
TNYMMX  .BYTE TZ,1,16.,10.,12.,15.,17.,20.,22.,20.,24.,27.,29.,27.,24.,26.,28.,30.,
        .BYTE TA,17.,26.,20.,1
        .BYTE T1,27.,39.,27.
        .BYTE TA,40.,48.,29.,1
        .BYTE TA,49.,64.,31.,1
        .BYTE TA,65.,80.,35.,1
        .BYTE TA,81.,99.,43.,1
TWTTFRA         .BYTE T1,1,20.,2
        .BYTE T1,21.,32.,2
        .BYTE T1,33.,99.,3
TWPULF  .BYTE T1,60.,99.,ZFIRYE
        .BYTE TE
 SEQUENCE CIRCLE,SQUARE,CROSS,PEANUT,KEY,TRIANGLE,CLOVER,V,STAIRS,U,FLAT,
        HEART,STAR,WAVES,TOPO,8
CAMWAV  .BYTE TZANDF,1,99.
        .BYTE NOJUMP-CAM
        .BYTE MOVJMP-CAM
        .BYTE SPIRAL-CAM
        .BYTE SPIRCH-CAM
        .BYTE COWJM2-CAM
        .BYTE MOVJMP-CAM
        .BYTE SPIRCH-CAM
        .BYTE SPIRAL-CAM
        .BYTE COWJM2-CAM
        .BYTE AVOIDR-CAM
```

```
          .BYTE SPIRCH-CAM
          .BYTE SPIRAL-CAM
          .BYTE COWJM2-CAM
          .BYTE NOJUMP-CAM
          .BYTE AVOIDR-CAM
          .BYTE SPIRCH-CAM
WTABLE
          .WORD TWPULF,WPULFI
          .WORD TWTTFRA,WTTFRA
          .WORD TCHARFR,WCHARFR              INVADER S FIRE TIMER  FRAMES
          .WORD TCHAMX,WCHAMX                MAX # ENEMY SHOTS -1
          .WORD WFLIMI,WFLMIN                MIN # FLIPPERS
          .WORD WFLIMX,WFLMAX                MAX
          .WORD WPULMI,WPUMIN
          .WORD WPULMX,WPUMAX
          .WORD WTANMI,WTAMIN
          .WORD WTANMX,WTAMAX
          .WORD WSPIMI,WSPMIN
          .WORD WSPIMX,WSPMAX
          .WORD WFUSMI,WFUMIN
          .WORD WFUSMX,WFUMAX
          .WORD WPULPOT,PULPOT
          .WORD WPULTIM,PULTIM
          .WORD WWTAC2,WTACAR+2
          .WORD WWTAC3,WTACAR+3
          .WORD TINVMX,WINVMX
          .WORD TNYMMX,NWNYMC
          .WORD TELIHI,NWTELI
          .WORD TPUCHDE,PUCHDE      PULSAR CHASE DELAY
          .WORD CAMWAV,WFLICAM               FLIPPER CAM
          .WORD TSPIIN,WINVIL+ZABTRA
          .WORD TCHARIN,WCHARL
          .WORD TINVIN,WINVIL
          .WORD TWFUSC,WFUSCH
          .WORD TFUFRQ,WFUFRQ
WTABEND
          .PAGE
          .SBTTL  PARAMETER TYPE CODE EXTRACTION VECTORS
 INPUT Y POINTER TO 1ST PARAMETER IN RECORD
          TYPCOD RECORD TYPE
DOTYPE
          LDX TYPCOD
          LDA X,SPARAD+1
          PHA
          LDA X,SPARAD
          PHA
          RTS
 INPUT Y PTS. TO END RANGE FIELD
DONEXT
          LDX TYPCOD
          LDA X,NPARAD+1
          PHA
          LDA X,NPARAD
          PHA
          RTS
SPARAD    .WORD 0                            EOT
          .WORD SAMALL-1                     ONE BYTE FOR ALL
          .WORD ITMIZE-1                     ITEMIZED BYTE/LEVEL
          .WORD DOTZAN-1
          .WORD DOTA-1
```

```
          .WORD DOTB-1
          .WORD DOTR-1
NPARAD    .WORD 0
          .WORD ONEBYT-1
          .WORD NITMIZ-1
          .WORD NITMIZ-1
          .WORD TWOBYT-1
          .WORD ONEBYT-1
          .WORD TWOBYT-1
DOTZAN    LDA TEMP2
          SEC
          SBC I,1
          AND I,0F
          CLC
          ADC I,1
          BPL ITMIZ2
ITMIZE    LDA TEMP2
ITMIZ2    STY TEMP0                          ITEMIZED BYTE FOR EACH WAVE
          DEY
          DEY
          SEC
          SBC NY,TEMP3
          CLC
          ADC TEMP0
          TAY
SAMALL                                       SAME BYTE FOR EACH WAVE IN RANGE
          LDA NY,TEMP3
          RTS
TWOBYT    INY
ONEBYT    INY
          INY
          RTS
NITMIZ    LDA NY,TEMP3
          DEY
          SEC
          SBC NY,TEMP3
          STA TEMP0
          TYA
          SEC
          ADC TEMP0
          TAY
          INY
          INY
          RTS


DOTB      LDA NY,TEMP3
          CLC
          ADC WINVIL
          RTS

DOTA
          JSR RANGER
          TAX
          LDA NY,TEMP3
          INY
          CPX I,0
          IFNE
          BEGIN
```

```
        CLC
        ADC NY,TEMP3
        DEX
        EQEND
        ENDIF
        RTS
RANGER  LDA TEMP2               CALCULATE # OF LEVELS BETWEEN
        STY TEMP0                       START AND END INCLUSIVE  ACC .
        DEY                             PRESERVE Y
        DEY
        SEC
        SBC NY,TEMP3
        INY
        INY
        RTS

DOTR                                    ALTERNATE BETWEEN 2 VALUES
        JSR RANGER
        AND I,1
        IFNE
        INY
        ENDIF
        LDA NY,TEMP3
        RTS
        .PAGE
        .SBTTL PLAY - MAINLINE  TOP OF WELL
PLAY
        JSR MOVCUR              MOVE CURSOR AROUND
        JSR FIREPC              FIRE PLAYER CHARGE
        JSR PROSUZ              PROCESS SUPER ZAP
        JSR MOVNYM              MOVE NYMPHS
        JSR MOVINV              MOVE INVADERS
        JSR MOVCHA              MOVE CHARGES
        JSR FIREIC              FIRE INVADER CHARGE
        JSR COLLIS              COLLISION DETECT
        JSR PROEXP              EXPLOSIONS
        JMP ANALYZ              ANALYZE PLAYER STATUS
        .PAGE
        .SBTTL  PLAY - MAINLINE  DROP MODE

                                PLAYER IS SHOOTING THRU TUBE TO GET TO NEXT

PLDROP
        LDA ELICNT              CLEAR WARNING REQUEST
        AND I,7F
        STA ELICNT
        JSR MOVCUR              MOVE CURSOR AROUND
        JSR MOVCUD              MOVE CURSOR DOWN
        JSR PROEXP              EXPLOSIONS
        JSR FIREPC              FIRE PLAYER CHARGES
        JSR MOVCHA              MOVE CHARGES
        LDA CURSL2
        IFMI                    CURSOR DEAD
        JSR ANALYZ              YES. ANALYZE CURSOR STATUS
        ENDIF
        RTS
        .PAGE
        .SBTTL PLAY - MOVE CURSOR  PRELIMINARY CHECK
MOVCUR
        LDA CURSL2
```

```
        IFMI                    -- CURSOR DEAD
        RTS                     YES. DON T MOVE IT
        ENDIF
        .SBTTL PLAY - MOVE CURSOR  MAINLINE
        LDX I,0
        LDA QSTATUS
        IFPL                    ATTRACT
        JSR AUTOCU              YES, AUTO MOVEMENT
        ELSE
        LDA TBHD                NO. MANUAL
        IFMI                    MAXIMIZE KNOB READING
        CMP I,-1F
        IFCC
        LDA I,-1F
        ENDIF
        ELSE
        CMP I,1F
        IFCS
        LDA I,1F
        ENDIF
        ENDIF
        STX TBHD
        ENDIF
        STA TEMP2
        EOR I,0FF               INVERT READING
        SEC
        ADC CURSPO              UPDATE CURSOR MASTER POSITION
        STA TEMP3               NEW CURSPO
        LDX WELTYP
        IFNE                    PLANAR SURFACE
        CMP I,0F0               YES.
        IFCS                    SPLIT CURSOR  WRAP
        LDA I,0EF               YES. MOVE AWAY FROM EDGE
        STA TEMP3
        ENDIF
        EOR TEMP2
        IFMI
        LDA TEMP3
        EOR CURSPO
        IFMI                    WRAPPED AROUND
        LDA CURSPO              YES.
        IFPL                    OLD POSITION LOW OR HI
        LDA I,00                LOW END

        ELSE
        LDA I,0EF               HIGH END

        ENDIF
        STA TEMP3               NEW CURSPO
        ENDIF
        ENDIF
        ENDIF
        LDA TEMP3               NEW CURSPO
        LSR
        LSR
        LSR
        LSR
        STA TEMP1               NEW CURSL1
        CLC
        ADC I,1                 CCW ADJACENT LINE # FOR CURSOR IS 1 AWAY
```

```
        AND I,0F
        STA TEMP2                   NEW CURSL2
        LDA TEMP1
        CMP CURSL1
        IFNE                        NEW POSITION
        JSR SBOING                  YES. MAKE SOUND
        ENDIF
        LDA TEMP1                   UPDATE CURSOR POSITION
        STA CURSL1
        LDA TEMP2
        STA CURSL2
        LDA TEMP3
        STA CURSPO
        RTS
        .SBTTL  PLAY-AUTO MOVE OF CURSOR
AUTOCU  LDA I,-1
        STA TEMP0
        STA TEMP1
        LDX WINVMX
        BEGIN                       LOOP FOR ALL INVADERS
        LDA X,INVAY
        IFNE                        ALIVE
        CMP TEMP0                   YES.
        IFCC                        HIGHEST
        STA TEMP0                   YES.
        STX TEMP1
        ENDIF
        ENDIF
        DEX
        MIEND
        LDX TEMP1
        IFPL
        LDA X,INVAL1
        LDY CURSL1
        JSR POLDEL                  HOW FAR   BEST DIRECTION
        TAY
        IFNE                        ALREADY THERE
        IFPL                        YES. WHICH WAY
        LDA I,-9
        ELSE
        LDA I,9
        ENDIF
        ENDIF
        ENDIF
        RTS

        .SBTTL  PLAY-MOVE CURSOR DOWN
MOVCUD  LDA CURSL2
        IFMI
        RTS
        ENDIF
        LDA CURMOD
        IFPL                        CURSOR DROPPING
        RTS                         NO
        ENDIF
                                    YES.
        LDA CURSY
        CMP I,ILINLI
        IFEQ                        STILL AT TOP
        JSR SOUTS2                  YES. START RUMBLE
```

```
        ENDIF
        LDA CURSYL              UPDATE CURSOR DEPTH
        CLC
        ADC CURSVL
        STA CURSYL
        LDA CURSY
        ADC CURSVH
        STA CURSY
        IFCC
        CMP I,ILINDDY
        ENDIF
        IFCS                    IS CURSOR PAST BOTTOM
        LDA I,CENDWA            YES. INITIALIZE SPACE MODE
        STA QSTATE
        JSR SOUTS3              START SPACE SOUND
        LDA I,OFF
        STA CURSY
        ENDIF
        LDA CURSY
        CMP I,50
        IFCS
        LDA PLAGRO
        IFEQ
        JSR INSTAR
        ENDIF
        ENDIF
        LDA EYLL                UPDATE EYE POSITION
        CLC
        ADC CURSVL
        STA EYLL
        LDA EYL
        ADC CURSVH
        IFCS
        INC EYH
        ENDIF
        CMP EYL
        IFNE                    EYE POSITION CHANGE
        INC ROTDIS              YES. REQUEST NEW WELL DISPLAY
        ENDIF
        STA EYL
                                CONSTANT ACCELERATION FOR VELOCITY
        LDA CURWAV              WAVE ACCELERATION +
        ASL
        ASL
        CMP I,30
        IFCS                    MAX OUT
        LDA I,30
        ENDIF
        CLC
        ADC I,20                BASE ACCELERATION
        CLC
        ADC CURSVL
        STA CURSVL
        LDA CURSVH
        ADC I,0
        STA CURSVH

                                CHECK FOR COLLISION WITH ENEMY LINES

        LDA CURSY
```

```
        CMP I,ILINDDY
        IFCC
        LDX I,NLINES-1          CURSOR STILL ON LINES
        BEGIN                   LOOP FOR EACH LINE
        LDA X,LINEY
        IFNE                    ACTIVE LINE
        CPX CURSL1              YES.
        IFEQ                    SAME LINE AS CURSOR
        CMP CURSY               YES.
        IFCC                    CURSOR AT ENEMY LINE POSITION
        JSR PULSTO              TURN OFF THRUST SOUND
        JSR INPPSQ              YES. START BANG. KILL CURSOR
        LDA I,0                 TURN OFF STARFIELD, EXIT LOOP
        STA PLAGRO
        JSR INICHA              CLEAR OUT ALL CHARGES
        ENDIF
        ENDIF
        ENDIF
        DEX
        MIEND
        ENDIF
        RTS
        .PAGE
        .SBTTL PLAY - MOVE NYMPHS
MOVNYM
        LDY I,0
        STY NEOFLI              CLEAR NEW OFF LIMITS FLAGS
        LDA INMCOU
        CLC
        ADC INCCOU
        CMP WINVMX
        IFCS                    INVADER SLOTS BOOKED ALREADY
        IFNE
        LDY I,-1                YES.
        ENDIF
        ENDIF
        LDA SUZTIM              AVOID KAMIKAZE
        IFNE
        LDY I,-1
        ENDIF
        STY TEMPY               ALLOW/DISALLOW UP NYMPH MOVEMENT
        LDX I,NNYMPH-1
        BEGIN                   LOOP FOR EACH NYMPH
        LDA X,NYMPY
        IFNE                    ACTIVE
        BIT TEMPY               YES.
        IFPL                    UP MOVEMENT OK
        SEC                     YES.
        SBC I,1
        STA X,NYMPY
        IFEQ                    UPDATE NYMPH POSITION. CONVERT
        JSR CONYMP              YES. MAKE IT AN INVADER
        ELSE
        CMP I,3F                NO.
        IFEQ                    JUST ENTERING ALONE ZONE
        LDY X,NYMPL             YES.
        LDA NEOFLI
        ORA NEOFLI
        AND Y,D70MSK
        IFNE                    ALREADY OCCUPIED
```

```
        INC X,NYMPY             YES. BACK OFF
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        LDA X,NYMPY
        CMP I,40
        IFCS                    DON T ROTATE PAST A CERTAIN PT.
        LDA QFRAME              OK TO ROTATE
        AND I,1
        IFEQ                    TIME TO ROTATE
        LDA X,NYMPL             YES. ROTATE NYMPH
        CLC
        ADC I,1
        AND I,0F
        STA X,NYMPL
        ENDIF
        ELSE
        CMP I,20                NO ROTATE.
        IFCS                    IN ALONE ZONE
        LDY X,NYMPL             YES.
        LDA Y,D70MSK            MARK LINE OFF LIMITS
        ORA NEOFLI
        STA NEOFLI
        ENDIF
        ENDIF
        ENDIF
        DEX
        MIEND
        LDA NEOFLI
        STA OLOFLI              NEW TO OLD OFF LIMITS
        RTS

        .SBTTL  PLAY - CONVERT NYMPH TO INVADER

CONYMP
        LDA I,ILINDDY           START AT BOTTOM
        STA TEMPO
        LDA X,NYMPL             START LINE
        STA TEMP1
        STX SAVEX
        JSR NYMCHA              NYMPH CHARACTERISTICS
        LDX SAVEX
        LDA TEMPO
        IFNE
        JSR ACTINV              ACTIVATE AN INVADER
        IFNE                    SLOT FOUND
        DEC NYMCOU              YES. DECREMENT NYMPH COUNT
        LDA I,0
        STA X,NYMPY             DEACTIVATE INVADER
        RTS
        ENDIF
        ENDIF
        LDA I,0FF               NO. STOP UP MOVEMENT FLAG
        STA TEMPY
        INC X,NYMPY             MOVE NYMPH BACK TO OLD POSITION
        RTS
        .PAGE
        .SBTTL  PLAY - ACTIVATE INVADER
 INPUT TEMPO Y POSITION AT WHICH TO START INVADER
```

```
        TEMP2,3 CHARACTERISTICS OF NEW INVADER
        TEMP1 CW LINE #
        TEMP4 CAM VALUE

 OUTPUT IF A SLOT IS FOUND  INMCOU INCREMENTED
                            INVAC1,2 N  UPDATED WITH CHARACTERISTICS
                            INVAL1      CW LINE #
                            INVAL2      CCW LINE #
                            INVAY       Y POSITION
                            INVCAM      CAM PC
                            INVACT      0
                            STATUS FLAGS 0
         IF NO SLOT IS FOUND STATUS FLAGS 0
         X,Y PRESERVED
                 SAVEY DESTROYED
ACTINV
         STY SAVEY
         LDY WINVMX
         BEGIN                      LOOP THRU INVADERS UNTIL SLOT IS FOUND
         LDA Y,INVAY
         IFEQ                       SLOT
         LDA TEMP0                  YES.
         STA Y,INVAY                Y
         LDA TEMP1
         CMP I,0F
         IFEQ                       POTENTIAL PLANAR SPLIT
         BIT WELTYP                 YES
         IFMI                       PLANAR
         LDA RANDOM                         YES. NO SPLITS
         AND I,0E
         ENDIF
         ENDIF
         STA Y,INVAL1               CW LINE
         CLC
         ADC I,1
         AND I,0F
         STA Y,INVAL2               CCW LINE
         LDA I,0
         STA Y,INVACT               TIMER
         LDA TEMP3
         STA Y,INVAC2
         LDA TEMP4
         STA Y,INVCAM
         INC INMCOU                 INVADER COUNT
         LDA TEMP2
         STA Y,INVAC1               CHARACTERISTICS
         LDY SAVEY
         AND I,INVABI
         STX SAVEY
         TAX
         INC X,FLIPCO               UPDATE INVADER TYPE COUNTER
         LDX SAVEY                  RESTORE X
         LDA I,10                   SOT FOUND FLAG
         RTS
         ENDIF
         DEY
         MIEND
         LDY SAVEY
         LDA I,0                    SLOT NOT FOUND FLAG
         RTS
```

```
          .PAGE
          .SBTTL  PLAY - DETERMINE NYMPH TYPE
NYMCHA
          LDA I,0
          LDX I,4
          BEGIN
          STA X,OPFLIP              0 ALL OPENING COUNTERS
          DEX
          MIEND
          LDX I,4
          BEGIN                     LOOP FOR EACH TYPE-CHECK MAX
          LDA X,WFLMAX
          SEC
          SBC X,FLIPCO
          IFCS                      MAX OF TYPE ALREADY
          STA X,OPFLIP              NO SAVE # OPENINGS
          ENDIF
          DEX
          MIEND

                                    TAKE AWAY Z OPENINGS OF TYPE FOR EACH TANKER

          LDY WINVMX
          BEGIN                     LOOP FOR EACH INVADER
          LDA Y,INVAY
          IFNE                      ALIVE
          LDA Y,INVAC2              YES.
          AND I,INVCAR              CARRIER
          IFNE
          TAX                       YES.
          CPX I,ZCARFU
          IFEQ
          LDX I,ZABFUS+1
          ENDIF
          DEC X,OPFLIP-1            2 LESS OPENINGS OF THAT TYPE
          DEC X,OPFLIP-1
          ENDIF
          ENDIF
          DEY
          MIEND
          LDX I,4
          LDA WINVMX
          CLC
          ADC I,1
          BEGIN                     LOOP FOR EACH TYPE-CALC TOTAL # OPENINGS
          SEC
          SBC X,FLIPCO
          DEX
          MIEND
          LDX I,4
          BEGIN                     LOOP FOR EACH TYPE
          CMP X,OPFLIP
          IFCC                      IF TOTAL # OPENINGS  TYPE OPENINGS
          STA X,OPFLIP              THEN DECREASE TYPE OPENINGS
          ENDIF
          DEX
          MIEND
          LDX I,4
          LDY I,0
          BEGIN                     LOOP FOR EACH TYPE
```

```
        LDA X,OPFLIP
        IFNE
        INY                     COUNT # TYPES WITH OPENINGS
        ENDIF
        DEX
        MIEND
        TYA
        IFNE                    OPENING
        DEY                     YES.
        IFEQ                    ONLY 1 TYPE
        LDX I,4                 YES.
        BEGIN                   LOOP UNTIL THAT ONE IS FOUND
        LDA X,OPFLIP
        IFNE
        LDA X,WFLMIN            YES
        IFNE                    LAUNCH OK
        JSR NEWTYP              NO. TRY FOR TYPE
        IFNE                    GOT IT
        RTS                     YES. EXIT
        ENDIF                   NO. KEEP TRYING
        ENDIF
        ENDIF
        DEX
        MIEND
        ELSE
        STY SXL                 NO.
        LDX I,4
        BEGIN                   LOOP FOR EACH TYPE-CHECK MINS
        LDA X,OPFLIP
        IFNE                    TYPE OPENINGS
        LDA X,FLIPCO            YES.
        CMP X,WFLMIN
        IFCC                    TYPE MIN SATISFIED
        JSR NEWTYP              NO. TRY FOR TYPE
        IFNE                    GOT IT
        RTS                     YES. EXIT
        ENDIF                   NO. KEEP TRYING
        ENDIF
        ENDIF
        DEX
        MIEND
                                MINS ARE OK.
        LDA OPSPIN              TRY FOR SMART LAUNCH
        IFNE
        LDA OPTANK
        IFNE                    SLOTS FOR TANKERS   SPINNER OPEN
        LDY TEMP1               YES.
        LDA Y,LINEY
        IFEQ                    LINE DEAD
        LDA I,OFF               YES. REAL SHORT THEN
        ENDIF
        LDX I,OPSPIN-OPFLIP     SHORT LINE LAUNCH SPINNER
        CMP I,OCC
        IFCC                    LONG ENEMY LINE
        LDX I,OPTANK-OPFLIP     YES LAUNCH TANKER
        ENDIF
        JSR NEWTYP              NO. TRY FOR TYPE
        IFNE                    GOT IT
        RTS                     YES. EXIT
        ENDIF                   NO. KEEP TRYING
```

```
        ENDIF
        ENDIF
        LDA RANDO2                  RANDOM TYPE  ELIM TYPE 0 THO
        AND I,3
        TAX
        INX
        LDY I,4                     START AT RANDOM SPOT AND
        BEGIN                       LOOP UNTIL NEEDY TYPE FOUND
        LDA X,WFLMIN
        IFNE                        OK FROM BOTTOM  NOT 0
        LDA X,OPFLIP                NO.
        IFNE                        NEEDY TYPE
        JSR NEWTYP                  YES. TRY LAUNCH
        IFNE                        GOT IT
        RTS                         YES. EXIT
        ENDIF
        ENDIF
        ENDIF
        DEX
        IFMI
        LDX I,4                     WRAP
        ENDIF
        DEY
        MIEND
        ENDIF
        ENDIF
        LDA I,0                     SIGNAL FAILURE
        STA TEMPO
        RTS
NEWTYP
        TXA
NEWTY2  ASL
        TAY
        LDA Y,NYMTAD+1
        PHA
        LDA Y,NYMTAD
        PHA
        RTS

NYMTAD  .WORD NEWFLI-1
        .WORD NEWPUL-1
        .WORD NEWTAN-1
        .WORD NEWSPI-1
        .WORD NEWFUS-1
NEWFLI
        LDA TNEWI2+ZABFLI
        STA TEMP3                   INVAC2
        LDA WFLICAM
        LDY I,ZABFLI                FLIPPER  INVAC1
        BEQ NEWGN3                  ALWAYS
NEWPUL                      PULSAR
        LDA TNEWI2+ZABPUL
        ORA WPULFI                  PULSAR FIRE
        LDY I,ZABPUL                PULSAR  INVAC1
        BNE NEWGN2                  ALWAYS
NEWFUS                              FUSE
        LDY I,ZABFUS
        BNE NEWGEN
NEWSPI                              SPINNER
        LDY I,ZABTRA
```

```
         BNE NEWGEN
NEWTAN                           TANKER
         LDA RANDOM
         AND I,3
         TAY
         LDA I,4
         STA TEMP2
         STX INDEX3              SAVE X
         BEGIN                   LOOP 4 TIMES  FAIL  OR UNTIL OPENING
         DEC TEMP2
         IFMI                    FAILURE FOR ALL
         LDX INDEX3              YES. RESTORE X
         LDA I,0                 SIGNAL FAILURE
         RTS
         ENDIF
         DEY
         IFMI                    CYCLE BETWEEN 0+3
         LDY I,3
         ENDIF
         LDX Y,WTACAR            GET TYPE OF TANKER
         CPX I,ZCARFU
         IFEQ
         LDX I,ZABFUS+1
         ENDIF
         LDA X,OPFLIP-1
         NEEND                   EXIT IF OPENINGS FOR TYPE
         LDX INDEX3              RESTORE X
         LDA Y,WTACAR            GET TANKER CONTENTS
         ORA I,ZFIRYE
         LDY I,ZABTAN
         BNE NEWGN2
NEWGEN
         LDA Y,TNEWI2
NEWGN2   STA TEMP3
         LDA Y,TNEWCAM
NEWGN3   STY TEMP2               GENERAL
         STA TEMP4
         LDA TEMP0               SUCCESS SIGNAL
         RTS
TNEWCAM          .BYTE NOJUMP-CAM,PULSCH-CAM,NOJUMP-CAM
         .BYTE TRALUP-CAM,FUSEUP-CAM
TNEWI2   .BYTE ZCARNO ZFIRYE ZDIRUP
         .BYTE ZCARNO ZFIRNO ZDIRUP
         .BYTE ZCARFL ZFIRYE ZDIRUP
         .BYTE ZCARNO ZFIRYE ZDIRUP
         .BYTE ZCARNO ZFIRNO ZDIRUP
         .SBTTL  PLAY - DETERMINE SPLIT INVADER CHARACTERISTICS
 INPUT   Y INVADER INDEX
 TEMP2 INVABI TYPE CODE          TEMP0 SPLIT DEPTH

SPLCHA   STY SAVEY
         LDA TEMP0
         CMP I,20
         LDA TEMP2
         IFCC                    SPLITTING TOO CLOSE TO PLAYER
         TAY                     YES. NO FLIPPING
         JSR NEWGEN
         ELSE
         JSR NEWTY2              NO. ASSIGN NORMAL PARAMETERS
         ENDIF
```

```
        LDY SAVEY
        RTS
        .PAGE
        .SBTTL PLAY - MOVE INVADERS  MAINLINE
MOVINV  LDA CURSL2
        IFPL                        PLAYER DEAD OR DROPPING
                                    YES. EXIT

        LDX WINVMX
        STX INDEX1
        BEGIN                       LOOP FOR EACH INVADER
        LDX INDEX1
        LDA X,INVAY
        IFNE                        ACTIVE
        LDA I,1                     SET NO EXIT FLAG
        STA EXICAM
        LDA X,INVCAM                SET UP INVADER S CAM PC
        STA CAMPC
        BEGIN                       LOOP UNTIL EXIT REQUESTED
        LDA CAMPC
        TAY                         GET INTO INTO CAM TABLE
        LDA Y,CAM                   GET CAM CODE
        JSR JSRCAM                  EXECUTE CAM REQUESTED
        INC CAMPC                   AUTO INCREMENT CAM PC
        LDA EXICAM                  EXIT REQUESTED
        EQEND
        LDA CAMPC                   UPDATE INVADER S CAM PC
        STA X,INVCAM
        ENDIF
        DEC INDEX1
        MIEND
        ENDIF
                                    UPDATE PULSE STATUS
        LDA PULSON
        CLC
        ADC PULTIM
        TAY
        EOR PULSON
        STY PULSON
        IFMI                        PULSAR STATUS CHANGE
        TYA                         YES.
        IFMI                        GO OFF
        JSR PULSTO                  YES. TURN OFF
        ELSE
        LDA FLIPCO+ZABPUL           NO. TURN ON IF ACTIVE PULSARS
        IFNE
        LDA CURSL2
        IFPL
        JSR PULSTR                  ACTIVE SO TURN ON
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        LDA PULSON
        IFPL                        BONUCE BETWEEN-27. AND +15.
        CMP I,15.
        BCS NEGPUL
        ELSE
        CMP I,-63.
        IFCC
```

```
NEGPUL  LDA PULTIM              NEGATE INCREMENT
        EOR I,0FF
        CLC
        ADC I,1
        STA PULTIM
        ENDIF
        ENDIF
        RTS


        .SBTTL  PLAY - INVADERS - CAM DISPATCHER

JSRCAM  TAY                     JSR INDIRECT TO CAM ROUTINE
        LDA Y,TABJSR+1
        PHA
        LDA Y,TABJSR
        PHA
        RTS
        .SBTTL  CAM TABLE MACROS
        .MACRO CAMAC ...X,...Y,...W
        .WORD ...X-1
        .MACRO ...Y
        .BYTE ...W
        .ENDM
        .ENDM

        .MACRO CAMA2I ...X,...Y,...W
        .WORD ...X-1
        .MACRO ...Y,...Z
        .BYTE ...W
        .BYTE ...Z
        .ENDM
        .ENDM

        .MACRO CAMA2F ...X,...Y,...W
        .WORD ...X-1
        .MACRO ...Y,...Z
        .BYTE ...W
        .BYTE ...Z-CAM-1
        .ENDM
        .ENDM
        .MACRO TEST1 X
        .BYTE X
        .ENDM
        .SBTTL  CAM TABLE SUBROUTINE POINTERS

TABJSR
        CAMAC JEXIT,VEXIT,0
        CAMA2I JSLOOP,VSLOOP,2
        CAMAC JSKIP0,VSKIP0,4
        CAMA2F JSETPC,VSETPC,6
        CAMA2F JELOOP,VELOOP,8
        CAMAC JNOOP,VNOOP,0A
        CAMAC JSMOVE,VSMOVE,0C
        CAMAC JSTRAI,VSTRAI,0E
        CAMA2I JSLOPB,VSLOPB,10
        CAMAC JJUMPS,VJUMPS,12
        CAMAC JJUMPM,VJUMPM,14
        CAMAC JCHROT,VCHROT,16
        CAMAC JKITST,VKITST,18
```

```
          CAMA2F JBROPC,VBROPC,1A
          CAMAC JELTST,VELTST,1C
CAMAC     JFUSEUP,VSFUSE,1E
CAMAC     JFUSKI,VFUSKI,20
CAMAC     JPULMO,VSPUMO,22
CAMAC     JCHPLA,VCHPLA,24
CAMAC     JCHKPU,VCHKPU,26
TABJSE
          .PAGE
          .SBTTL  PLAY - INVADERS - CAM SUBROUTINES

                                    EXIT CAM

JEXIT
          LDA I,0                   EXIT CAM FLAG
          STA EXICAM
JNOOP     RTS

                                    SET CAM LOOP COUNTER

JSLOOP    INC CAMPC
          LDY CAMPC
          LDA Y,CAM
          STA X,INVLOO              NEW LOOP VALUE
          RTS
JSLOPB                      SAME AS JSLOOP EXCEPT OPERAND BP ADDRESS OF PARAMETER
          INC CAMPC
          LDY CAMPC
          LDA Y,CAM
          TAY                       BP LOC OF VALUE
          LDA Y,0                   NEW LOOP VALUE
          STA X,INVLOO
          RTS

                                    SKIP NEXT CAM LINE IF CAMSTA 0
                                     DOUBLE BYTE
JSKIP0    LDA CAMSTA
          IFEQ
          INC CAMPC
          INC CAMPC
          ENDIF
          RTS

           BRANCH IF CAMSTA 0

JBROPC    INC CAMPC
          LDA CAMSTA
          IFEQ                      BRANCH
          LDY CAMPC                 YES. SET NEW PC
          LDA Y,CAM
          STA CAMPC
          ENDIF
          RTS

                                    DEC LOOP VALUE
                                    IF 0 THEN EXIT
                                         ELSE RELOOP
JELOOP
          DEC X,INVLOO
          IFEQ
```

```
          INC CAMPC                   EXIT LOOP
          ELSE
JSETPC    LDY CAMPC                   NEW CAM PC
          LDA Y,CAM+1                 RELOOP
          STA CAMPC
          ENDIF
          RTS
JELTST
          LDY X,INVAL1
          LDA Y,LINEY
          IFEQ
          LDA I,0FF                   WORST CASE LINE  DEAD
          ENDIF
          CMP X,INVAY
          IFCC                        ENEMY ON AN ENEMY LINE
          LDA I,0                     YES.
          ELSE
          LDA I,1                     NO.
          ENDIF
          STA CAMSTA
          RTS

          .SBTTL  PLAY - INVADERS - CAM ROUTINES
JCHKPU                  CHECK FOR PULSING NOW OR IN NEXT 4 FRAMES
          LDA PULTIM
          ASL
          ASL
          CLC
          ADC PULSON
          AND PULSON
          AND I,80
          EOR I,80
          STA CAMSTA                  EXIT  0 NO PULSE  80 PULSE
          RTS
                                      CHANGE DIRECTION OF JUMP
JCHROT
          LDA X,INVAC1
          EOR I,INVROT
          STA X,INVAC1
          RTS

          .SBTTL PLAY - MOVE INVADERS  MOVE 1 UP
 INPUT    X INVADER INDEX

JSMOVE
          LDA X,INVAC1
          AND I,INVABI
          TAY                         INVADER TYPE
          LDA X,INVAC2
          IFPL                        GOING UP
JSMOVU    LDA X,INVAYL                YES.
          CLC
          ADC Y,WINVIL
          STA X,INVAYL                MOVE UP
          LDA X,INVAY
          ADC Y,WINVIN
          STA X,INVAY
          CMP CURSY
          BEQ ATOP
```

```
          IFCC                      AT TOP
ATOP      JSR CHASER                YES. CONVERT TO CHASER
          ELSE
          CMP I,20                  NO
          IFCC                      TOO CLOSE TO TOP FOR CARRIER
          LDA X,INVAC2              YES.
          AND I,INVCAR              CARRIER
          IFNE
          TXA                       YES.
          PHA                       SAVE X
          TAY
          JSR KILINV                SPLIT CARRIER
          PLA
          TAX                       RESTORE X
          ENDIF
          ENDIF
          ENDIF
          ELSE
JSMOVD    LDA X,INVAYL              DOWN
          SEC
          SBC Y,WINVIL
          STA X,INVAYL
          LDA X,INVAY
          SBC Y,WINVIN
          STA X,INVAY
          CMP I,ILINDDY
          IFCS                      AT BOTTOM
          LDA I,0F2
          STA X,INVAY               YES.
          ENDIF
          ENDIF
          RTS
          .PAGE
          .SBTTL  PLAY - INVADERS   PULSE MOVE
JPULMO    LDY I,ZABPUL
          LDA X,INVAC2
          IFPL                      GOING UP
          LDA X,INVAY               YES.
          CMP PULPOT
          IFCS                      IN POWER ZONE
          LDY I,ZABFLI              NO. GO FASTER
          ENDIF
          JSR JSMOVU                MOVE UP
          ELSE
          JSR JSMOVD                MOVE DOWN  RETURN WITH ACC Y POS
          LDY NYMCOU
          IFEQ                      NYMPHS GONE
          LDA I,0FF                 SEND PULSAR UP
          ENDIF
          CMP PULPOT
          IFCS                      TIME TO REVERSE
          LDA X,INVAC2              YES
          EOR I,INVDIR
          STA X,INVAC2
          ENDIF
          ENDIF
          LDA PULSON                YES. SEE IF CURSOR GOT ZAPPED
          IFPL                      PULSAR ON
          LDA X,INVAY               YES.
          CMP PULPOT
```

```
        IFCC                    PULSAR IN RANGE
        LDA CURSL1              YES
        CMP X,INVAL1
        IFEQ
        LDA CURSL2
        CMP X,INVAL2
        IFEQ                    ON CURSOR LINES
        JSR INPPSQ              YES. KILL CURSOR
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        RTS
CHKSM3          .BYTE QCHKS3
        .PAGE
        .SBTTL PLAY - INVADERS  CONVERT TO CHASER
 INPUT  X INVADER INDEX

CHASER
        LDA CURSY               PLACE EXACTLY AT TOP
        STA X,INVAY
        LDA X,INVAC1
        AND I,INVABI
        CMP I,ZABPUL
        IFEQ                    PULSAR
        LDA NYMCOU              YES.
        IFNE                    ANY MORE NYMPHS
        LDA X,INVAC2            YES. SEND IT DOWN
        EOR I,INVDIR
        STA X,INVAC2
        RTS                     EXIT
        ENDIF
        ENDIF
        LDA X,INVAC1
        IFMI                    STILL FLIPPING 2
        INC X,INVAY             YES. FINISH FLIP
        RTS                     BEFORE AT TOP STATUS
        ENDIF
        DEC INMCOU              -1 TO # WALL INVADERS
        LDA INCCOU
        CMP I,1
        IFNE                    OTHER THAN 1 CHASER
        JSR JCHPLA              YES. SEND CHASER SHORTEST WAY
        ELSE
                                NO. 1 OTHER CHASER, SO SEND
        LDY I,NINVAD-1          THIS GUY IN OPPOSITE DIRECTION
        BEGIN                   LOOP UNTIL OTHER CHASER IS FOUND
        LDA Y,INVAY
        IFNE
        STY INDEX2
        CPX INDEX2
        IFNE                    MAKE SURE IT S NOT NEW CHASER
        LDA Y,INVAY
        CMP CURSY
        BEQ GOTCHA              EXIT LOOP IF FOUND
        ENDIF
        ENDIF
        DEY
        MIEND
GOTCHA  LDA Y,INVAC1
```

```
        AND I,INVROT            GET OTHER CHASER S DIRECTION
        EOR I,INVROT            USE ITS OPPOSITS
                                SET CHASE DIRECTION
        STA X,INVAC1
        ENDIF
        LDA I,TOPPER-CAM-1
        STA CAMPC               SET CHASER CAM
        INC INCCOU              +1 TO CHASER COUNT
        RTS
        .SBTTL
JCHPLA
        LDA X,INVAL1            SEND CHASER SHORTEST WAY
        TAY
        LDA CURSL1
        JSR POLDEL              DETERMINE POLAR DELTA TO CURSOR
        ASL
        LDA X,INVAC1
        IFCC                    SET CHASE DIRECTION  SHORTEST WAY
        ORA I,INVROT            CCW
        ELSE
        AND I, CINVROT          CW
        ENDIF
        STA X,INVAC1
        RTS
        .PAGE
        .SBTTL PLAY - MOVE INVADERS  PROCESS JUMP
JJUMPM
                                UPDATE JUMP ANGLE
        LDY X,INVAL2
        LDA X,INVAC1
        AND I,INVROT
        IFEQ                    MOVING
        INY                     CW  JUMP ROTATION CCW
        ELSE
        DEY                     CCW  JUMP ROTATION CW
        ENDIF
        TYA                     NEW JUMP ANGLE
        AND I,0F                MOD 16
        ORA I,80                JUMP CODE
        STA X,INVAL2            UPDATED JUMP ANGLE
        LDA X,INVAC1
        AND I,INVABI
        CMP I,ZABFUS            FUSE AT A JUNCTION  IFEQ
        IFEQ
        LDA X,INVAL2            MAYBE.
        AND I,7
        IFEQ                    AT A JUNCTION
        LDA X,INVAL2            YES
        AND I,8
        IFNE                    MOVING CCW
        LDA X,INVAL1            YES. ADJUST BASE
        CLC
        ADC I,1
        AND I,0F
        STA X,INVAL1
        ENDIF
        LDA X,INVAC1            YES
        AND I, CINVMOT
```

```
        STA X,INVAC1            SET STATUS BACK TO LINE
        LDA I,020
        STA X,INVAL2            MAKE IT INVINCIBLE
        LDA X,INVAC2
        EOR I,INVDIR
        STA X,INVAC2            REVERSE UP DOWN DIRECTION
        LDA NYMCOU
        IFEQ                    NYMPHS GONE
        LDA X,INVAY             YES
        CMP CURSY
        IFEQ                    AT TOP
        JSR FUCHPL              YES. STAY THERE   CHASE PLAYER
        ELSE
        LDA X,INVAC2            NO. SEND UP.
        AND I,INVDIR
        STA X,INVAC2
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        ELSE
                                CALCULATE FINAL JUMP ANGLE
        LDY X,INVAL1
        LDA X,INVAC1
        EOR I,INVROT            BACKWARDS
        JSR CALSAN
        CMP X,INVAL2
        IFEQ                    FINAL JUMP ANGLE UPDATED ANGLE
        LDA X,INVAC1            YES
        AND I, C INVMOT
        STA X,INVAC1            SET STATUS BACK TO MOVER
        AND I,INVROT
        IFEQ                    NEW LINE IN WHICH DIRECTION
        LDA X,INVAL1            CW
        STA X,INVAL2
        SEC
        SBC I,1
        AND I,0F
        STA X,INVAL1
        ELSE
        LDA X,INVAL1            CCW
        CLC
        ADC I,1
        AND I,0F
        STA X,INVAL2
        ENDIF
        ENDIF
        ENDIF
        LDA X,INVAC1
        AND I,INVMOT            RETURN WITH STATUS  0 JUMP DONE
        STA CAMSTA              SET CAM STATUS
        RTS
        .PAGE
        .SBTTL PLAY - MOVE INVADERS  CHASE PLAYER

JKITST
        LDA X,INVAC1
        IFPL                    MOVING  NOT JUMPING
        LDA X,INVAL1            YES
        CMP CURSL1
        IFEQ                    IS ANY INVADER LEG ON SAME LINE
```

```
        LDA X,INVAL2            AS ANY CURSOR LEG
        CMP CURSL2
        IFEQ
        JSR INIPSQ              YES. DESTROY CURSOR
        ENDIF
        ENDIF
        ENDIF
        RTS
JFUSKI  LDA X,INVAY             CHECK FOR FUSE KILL CURSOR
        CMP CURSY
        IFEQ                    SAME HEIGHT
        LDA X,INVAL1            YES.
        CMP CURSL1
        IFEQ                    SAME LINE
        JSR INFPSQ              YES. DEAD CURSOR NOW
        ENDIF
        ENDIF
        RTS

        .SBTTL PLAY - MOVE INVADERS  START A JUMP
 INPUT  INVACO X  BIT IJDIRE
        ACC BIT IJDIRE JUMP DIRECTION  JUMPSD ENTRY ONLY
 OUTPUT INVACO X  BIT IMOVER
        INVAL1 X  SET TO BASE LEG
        INVAL2 X  SET TO JUMP SEQ START

JJUMPS
        JSR OKTOJM              VERIFY JUMP DIRECTION
JUMPSD
        LDA X,INVAC1
        ORA I,ZMOTJM
        STA X,INVAC1            SET JUMPS STATUS
        AND I,INVABI
        CMP I,ZABFUS
        IFEQ                    FUSE
        LDA X,INVAC1            YES.
        AND I,INVROT
        IFEQ                    WHICH WAY
        LDA I,81                CCW
        ELSE
        LDA X,INVAL1            CW
        SEC
        SBC I,1
        AND I,0F
        STA X,INVAL1
        LDA I,87
        ENDIF
        STA X,INVAL2
        ELSE
        LDA X,INVAC1            NO
        AND I,INVROT
        IFNE                    MOVING CCW
        LDA X,INVAL1            YES. ADJUST BASE LEG
        CLC
        ADC I,1
        AND I,0F
        STA X,INVAL1
        ENDIF
        LDA X,INVAC1            NO.
        LDY X,INVAL1
```

```
        JSR CALSAN              CALC. STARTING ANGLE
        STA X,INVAL2
        ENDIF
        RTS
OKTOJM  LDA WELTYP
        IFNE                    PLANAR SURFACE
        LDA X,INVAC1            YES
        AND I,INVROT
        IFNE                    MOVING CCW
        LDA X,INVAL1            CCW
        CMP I,0E
        IFCS                    AT RIGHT EDGE
        LDA X,INVAC1            YES CHANGE TO CW JUMP
        AND I, CINVROT
        STA X,INVAC1
        ENDIF
        ELSE
        LDA X,INVAL1            CW
        IFEQ                    AT LEFT EDGE
        LDA X,INVAC1            YES CHANGE TO CCW JUMPS
        ORA I,INVROT
        STA X,INVAC1
        ENDIF
        ENDIF
        ENDIF
        RTS

CALSAN                          CALCULATE STARTING JUMP ANGLE

                                BASE LEG IN Y
        AND I,INVROT
        IFNE                    MOVING CCW
        DEY                     YES.
        TYA
        AND I,0F
        TAY
        LDA Y,LINANG
        CLC                     YES. ADJUST ANGLE FOR BASE LEG ON
        ADC I,8                 RIGHT SIDE
        AND I,0F                MOD 16
        ELSE
        LDA Y,LINANG            CW
        ENDIF
        ORA I,80                JUMP CODE
        RTS
        .PAGE
        .SBTTL  PLAY-INVADER FUSE UP/DOWN MOTION

JFUSEUP         LDY I,ZABFUS
        LDA X,INVAC2
        IFPL                    UP OR DOWN
        LDA X,INVAYL            UP.
        CLC
        ADC WFUSIL
        STA X,INVAYL
        LDA X,INVAY
        ADC WFUSIH
        STA X,INVAY
        CMP CURSY
        IFCC                    AT TOP
```

```
        LDA CURSY                   YES
        STA X,INVAY
        ELSE
        LDY NYMCOU                  NO
        IFNE                        NYMPHS LEFT
        LDY CURWAV                  YES.
        CPY I,17.
        IFCC                        EARLY WAVE
        CMP I,20                    YES. TURN BACK BEFORE TOP
        ENDIF
        ELSE
        RTS                         NONE LEFT. HEAD FOR TOP
        ENDIF
        ENDIF
        IFCC                        TOO HIGH
        LDA WFUSCH                  YES.
        IFMI                        CHASE PLAYER AT TOP
        JSR FUCHPL                  YES. CHASE
        ELSE
        JSR LEFRIT                  NO. RANDOM
        ENDIF
        ELSE
        JSR MAYBLR                  NO. MAYBE GO LEFT OR RIGHT ANYWAY
        ENDIF
        ELSE
        JSR JSMOVD                  MOVE DOWN
        CMP I,080
        IFCS                        AT BOTTOM OF RANGE
        BIT WFUSCH                  YES.
        IFVS                        CHASE PLAYER ON TUBE
        JSR FUCHPL                  YES. CHASE
        ELSE
        JSR LEFRIT                  NO. RANDOM
        ENDIF
        ELSE
        JSR MAYBLR                  NO. MAYBE GO LEFT OR RIGHT
        ENDIF
        ENDIF
        RTS
        .SBTTL  INVADER FUSE JUMP DECISION

MAYBLR
        LDA X,INVAY
        AND I,20
        IFNE
        LDA RAND02
        CMP WFUFRQ
        IFCS
        BIT WFUSCH
        IFVS                        CHASE PLAYERS ON TUBE
        TXA                         YES. ONLY IF INDEX IS EVEN
        LSR
        BCC LEFRIT
        JSR FUCHPL                  YES. CHASE
        ELSE
        JSR LEFRIT                  NO. RANDOM
        ENDIF
        ENDIF
        ENDIF
        RTS
```

```
        .SBTTL  INVADER FUSE LEFT/RIGHT VECTOR
FUCHPL  JSR JCHPLA              CHASE PLAYER
        JSR JCHROT              REVERSE DIRECTION  FUSE IS BACKWARDS
        JMP GOTJUM
LEFRIT
        LDA X,INVAC1            RANDOMLY CHOOSE LEFT OR RIGHT
        AND I, CINVROT
        BIT RANDOM
        IFVS
        ORA I,INVROT
        ENDIF
        STA X,INVAC1
GOTJUM  LDA WELTYP
        IFNE                    PLANAR SURFACE
        LDA X,INVAC1            YES.
        AND I,INVROT
        IFEQ                    GOING CCW
        LDA X,INVAL1            YES.
        CMP I,0F
        BCS REVFLP              AT RIGHT EDGE
        ELSE                    NO.
        LDA X,INVAL1            NO. GOING CW
        IFEQ                    AT LEFT EDGE
REVFLP  LDA X,INVAC1            YES. GO BACK
        EOR I,INVROT
        STA X,INVAC1
        ENDIF
        ENDIF
        ENDIF
        LDA I,FUSELR-CAM
        STA CAMPC               PT TO LEFT RIGHT FUSE CAM
        JMP JUMPSD              GO START JUMP
        .PAGE
        .SBTTL  PLAY - INVADERS -TRAILER


                                SPECIAL TRAILER PROCESSING

JSTRAI
        LDA I,1
        STA CAMSTA
        LDY X,INVAL1
        LDA Y,LINEY
        IFEQ                    LINE VACANT
        LDA I,ILINDDY+1         YES. START LOW. 2
        STA Y,LINEY
        ENDIF
        LDA X,INVAY
        CMP Y,LINEY
        IFCC                    NEW ENEMY LINE
        STA Y,LINEY             YES.
        LDA I,80
        STA Y,LINSTA            REQUEST RECALC.
                                NO DESTRUCTION PICS
        ENDIF
        LDA X,INVAY
        CMP I,20
        IFCC                    MAX HEIGHT
        LDA X,INVAC2            YES.
        ORA I,ZDIRDO            SEND IT DOWN
```

```
        STA X,INVAC2
        LDA I,20                MAX HEIGHT
        STA X,INVAY
        ELSE
        CMP I,0F2               NO.
        IFCS                    MIN HEIGHT
        JSR ASTRAL              YES. REASSIGN, REVERSE
        LDA I,0F0               DON T LET IT GET TO LOW
        STA X,INVAY
        LDA NYMCOU              ANY NYMPHS, OR NON SPIKER TYPE CLIMBERS
        IFEQ
        LDA X,INVAC2
        AND I, C INVCAR         CONVERT IT TO TANKER
        ORA I,ZCARFL            CARRYING FLIPPERS
        STA X,INVAC2
        LDA X,INVAC1            LOOKS LIKE TANKER TOO
        AND I, C INVABI
        ORA I,ZABTAN
        STA X,INVAC1
        LDA I,0                 SET ZERO STATUS  CONVERTED TOO CARRIER
        STA CAMSTA
        ENDIF
        ENDIF
        ENDIF
        RTS
ASTRAL
        LDA I,0
        STA TEMP4
        LDA I,NLINES-1          LOOP LINE COUNTER
        STA OPSPIN

        LDA RAND02              START AT A RANDOM LINE
        AND I,0F
        TAY
        BEGIN                   LOOP FOR EACH LINE
        CPY I,0F
        IFEQ
        LDA WELTYP
        BNE SKIPIT              SKIP LINE IF PLANAR FAR RIGHT EDGE
        ENDIF
        LDA Y,LINEY
        IFEQ                    DEAD LINE
        LDA I,0FF               YES. WORST CASE
        ENDIF
        CMP TEMP4
        IFCS                    NEEDIEST LINE SO FAR
        STA TEMP4               YES. CONDITION
        STY TEMP0               LINE #
        ENDIF
SKIPIT  DEY
        IFMI
        LDY I,NLINES-1
        ENDIF
        DEC OPSPIN
        MIEND
        LDA TEMP0               REASSIGN TO NEW LINE
        STA X,INVAL1
        CLC
        ADC I,1
        AND I,0F
```

```
        STA X,INVAL2
        LDA X,INVAC2            SEND BACK UP
        AND I, C INVDIR
        STA X,INVAC2
        RTS

        .SBTTL  PLAY - KILL INVADER
 INPUT Y INVADER TO BE SPLIT
 OUTPUT ORIGINAL KILLED OFF
        UP TO 2 NEW ONES CREATED
        X IS PRESERVED

KILINV
        LDA Y,INVAY             SAVE Y
        STA TEMP0
        CMP CURSY
        IFEQ                    DECREMENT COUNTER
        LDA Y,INVAC1
        AND I,INVABI
        CMP I,ZABFUS
        BEQ MOVER               FUSE  BRANCH IF FUSE  OR CHASE
        DEC INCCOU              CHASER
        ELSE
MOVER   DEC INMCOU              MOVER
        ENDIF
        LDA I,0                 DEACTIVATE ENEMY
        STA Y,INVAY
        LDA Y,INVAC1
        AND I,INVABI
        STX SAVEX
        TAX
        DEC X,FLIPCO            UPDATE TYPE COUNTER
        LDX SAVEX
        LDA Y,INVAC2
        AND I,INVCAR
        IFNE                    SPLIT TYPE INVADER
        SEC                     YES
        SBC I,1
        CMP I,ZABTAN
        IFEQ                    TANKER
        LDA I,ZABFUS            YES. REALLY FUSE
        ENDIF
        STA TEMP2               RESULTANT MUTATION
        LDA Y,INVAL1            YES.
        SEC
        SBC I,1
        AND I,0F
        CMP I,0F                DON T ALLOW WRAPAROUND ON PLANE
        IFCS
        BIT WELTYP
        IFMI
        LDA I,0
        ENDIF
        ENDIF
        STA TEMP1               LINE # CW
                                Y
        JSR SPLCHA              CHARACTERISTICS
        LDA TEMP4               JUST IN CASE THE DEAD
        STA CAMPC               SLOT GETS USED
        DEC CAMPC
```

```
        LDA I,0                 SET EXIT FLAG
        STA EXICAM
        JSR ACTINV              ACTIVATE AN INVADER
        IFNE                    ANY SLOTS
        LDA TEMP1               YES
        CLC
        ADC I,2
        AND I,0F
        CMP I,0F
        IFEQ                    DON T ALLOW WRAP AROUND ON PLANE
        BIT WELTYP
        IFMI
        LDA I,0E
        ENDIF
        ENDIF
        STA TEMP1               LINE #CCW
        LDA TEMP2
        ORA I,ZROCCW
        STA TEMP2
        JSR ACTINV              ACTIVATE ANOTHER INVADER
        ENDIF
        ENDIF
        RTS
        .PAGE
        .SBTTL  PLAY - INVADER CAM TABLES

CAM

                                TRAILER MOVING UP

TRALUP
        VSMOVE                  MOVE UP
        VSTRAI                  PROCESS TRALER
        VBROPC NOJUMP           CONVERT TO CARRIER
        VEXIT                   EXIT
        VSETPC TRALUP           RELOOP

                                MOVING UP  NO JUMPS

NOJUMP
        VSMOVE                  MOVE UP
        VEXIT
        VSETPC NOJUMP           RELOOP

                                MOVE 3 TIMES, THEN JUMP
MOVJMP
        VSLOOP 8
MJLOP1  VSMOVE                  MOVE UP N FRAMES
        VEXIT
        VELOOP MJLOP1
        VJUMPS                  START JUMP
MJLOP5  VEXIT
        VJUMPM                  PROCESS JUMP
        VSKIP0                  SKIP IF JUMP IS DONE
        VSETPC MJLOP5
        VSETPC MOVJMP           JUMP IS DONE. RESTART SEQUENCE

                                SMOOTH UPWARD SPIRAL

SPIRAL
```

```
        VSMOVE
        VEXIT
        VJUMPS                  START JUMP
SPILOP  VEXIT
        VJUMPM                  PROCESS JUMP
        VSMOVE                  MOVE UP
        VSKIPO
        VSETPC SPILOP
        VSETPC SPIRAL           RESTART JUMP WHEN FINISHED

                                CHANGE JUMP DIRECTION EVERY N JUMPS

SPIRCH
        VSMOVE
        VEXIT
        VSLOOP 2                LOOP FOR N JUMPS
SPRLP1  VJUMPS                  START JUMP
SPRLP2  VEXIT
        VJUMPM                  CONTINUE JUMP
        VSMOVE                  MOVE UP
        VSKIPO                  JUMP DONE
        VSETPC SPRLP2           NO. CONTINUE JUMP
        VEXIT
        VELOOP SPRLP1           YES. NEW JUMP OR EXIT
        VCHROT                  CHANGE JUMP DIRECTION
        VSLOOP 3                LOOP FOR N JUMPS
SPRLP3  VJUMPS                  START JUMP
SPRLP4  VEXIT
        VJUMPM                  CONTINUE JUMP
        VSMOVE                  MOVE UP
        VSKIPO                  JUMP DONE
        VSETPC SPRLP4           NO. CONT JUMP
        VEXIT
        VELOOP SPRLP3           YES. NEW JUMP OR EXIT
        VCHROT
        VSETPC SPIRCH           START OVER

                                CHASE PLAYER AROUND TOP

TOPPER
        VSLOOP 4                WAIT IN CROUCH FOR N FRAMES

KICHEK  VKITST                  TEST FOR CURSOR KILL
        VEXIT
        VELOOP KICHEK
        VJUMPS                  START A JUMP

KJULP1  VEXIT
        VSLOPB WTTFRA
KJULP2  VJUMPM                  DOUBLE SPEED JUMP
        VBROPC TOPPER           SKIP IF JUMP IS DONE
        VELOOP KJULP2
        VSETPC KJULP1
                                ENEMY FLIPS   MOVES ON OPEN LINES, MOVES ON ENEMY
COWJM2  VEXIT
COWJMP  VSMOVE                  MOVE ENEMY
        VELTST                  ON AN ENEMY LINE
        VBROPC COWJM2           YES. CONTINUE UP ON LINE
        VJUMPS                  NO. START A JUMP
        VEXIT
```

```
          VSMOVE                    MOVE UP
COWJM3    VJUMPM                    PROCESS JUMP
          VBROPC COWJM2             JUMP DONE
          VEXIT
          VSETPC COWJM3             CONTINUE JUMP

                                    PULSAR




                                    FUSE UP/DOWN
FUSEUP
          VSFUSE                    PROCESS FUSE
          VFUSKI                    FUSE KILL CURSOR
          VEXIT                     EXIT
          VSETPC FUSEUP             RELOOP

FUSELR    VEXIT                     FUSE LEFT/RIGHT
          VSLOOP 3                  SLOWL
FUSLOP    VFUSKI                    CURSOR KILLED
          VEXIT
          VELOOP FUSLOP
          VJUMPM                    LEFT/RIGHT
          VBROPC FUSEUP             JUMP DONE
          VSETPC FUSELR             NO. CONTINUE JUMP

PULSCH
PULSCP                              PULSAR CHASER PLAYER
          VSLOPB PUCHDE
PULSC1    VSPUMO                    MOVE 1/8 OF TUBE BEFORE NEXT FLIP
          VEXIT
          VELOOP PULSC1
PULSC2    VCHKPU                    PULSING
          VBROPC PULSC3             BRANCH IF NOT
          VSPUMO                    PULSING, SO KEEP MOVING
          VEXIT
          VSETPC PULSC2             RECHECK FOR PULSE
PULSC3    VCHPLA                    SET FLIP DIRECTION TOWARD PLAYER
          VJUMPS                    START FLIP
PULSCJ    VEXIT
          VJUMPM                    CONTINUE FLIP
          VBROPC PULSCP             DONE
          VSETPC PULSCJ             NO

                                    AVOIDANCE FLIPPER

AVOIDR
          VCHPLA                    SET DIRECTION TOWARD PLAYER
          VCHROT                    REVERSE IT
          VJUMPS
AVOID1    VEXIT                     FLIP PROCESSING LOOP
          VSMOVE
          VJUMPM
          VSKIPO
          VSETPC AVOID1
          VSLOOP 4.
AVOID2    VEXIT                     FLIP DONE. MOVE UP LOOP
          VSMOVE
          VELOOP AVOID2
          VSETPC AVOIDR
```

```
        .SBTTL PLAY - MOVE CHARGES
MOVCHA
        LDX I,NPCHAR+NICHAR-1
        STX INDEX1
        BEGIN                       LOOP FOR EACH CHARGE
        LDX INDEX1
        LDA X,CHARY
        IFNE                        CHARGE ACTIVE
        CPX I,NPCHAR
        IFCC                        DETERMINE DIRECTION
                                    TOWARD INVADER
        ADC I,PCVELO
        LDY X,CHARCO
        IFNE                        CHARGE IN COLLISION W. LINE
        SEC                         YES. SLOW IT DOWN
        SBC I,4
        ENDIF
        STA X,CHARY
        JSR LIFECT
        LDA X,CHARY
        CMP I,ILINDDY
        IFCS                        AT END
        DEC CHACOU
        LDA I,0                     YES, DEACTIVATE
        STA X,CHARY
        ENDIF
        ELSE
        LDA X,CHARYL
        CLC                         TOWARD PLAYER
        ADC WCHARL
        STA X,CHARYL
        LDA X,CHARY
        ADC WCHARIN
        CMP CURSY
        IFCC                        AT TOP
        DEC ESHCOU
        JSR CHATOP                  YES. CHECK FOR COLLISION WITH CURSOR
        LDA I,0                     DEACTIVATE
        ENDIF
        STA X,CHARY
        ENDIF
        ENDIF
        DEC INDEX1
        MIEND
        RTS
CHATOP                  CHECK FOR CURSOR CHARGE COLLISION
        LDA CURSL1
        CMP X,CHARL1
        IFEQ
        LDA CURSL2                  SAME LINE AS CURSOR.
        IFPL                        CURSOR ALREADY DEAD
        JSR INCPSQ                  NO. KILL CURSOR
        LDA I,81                    SPECIAL BLASTED CODE
        STA CURSL2
        ENDIF
        ENDIF
        RTS
        .SBTTL  PLAY - CHARGE LINE COLLISION
```

```
LIFECT                          PROCESS PLAYER CHARGE S EFFECT
                                ON ENEMY LINES

        LDY X,CHARL1            DO CHARGE LINE 1 FIRST
        LDA Y,LINEY
        IFNE                    LINE DEAD
        LDA X,CHARY             NO.
        CMP Y,LINEY
        IFCS                    CHARGE ON ENEMY LINES
        CMP I,ILINDDY           YES
        IFCS                    LINE DEAD
        LDA I,0                 YES
        ENDIF
        STA Y,LINEY             YES. UPDATE LINE ENEMY TO
        INC X,CHARCO            UPDATE CHARGE - ENEMY LINE COLLISION COUNTER
        LDA I,0C0
        STA Y,LINSTA            SET RECALC FLAG
                                REQUEST LINE DESTRUCTION PIC.
        JSR SELICO              MAKE SOUND
                                GIVE PTS
        LDX I,-1                SIGNAL SCORE ROUTINE TO USE TEMPS
        LDA I,0                 ADD 1 TO SCORE FOR EACH HIT
        STA TEMP1
        STA TEMP2
        LDA I,1
        STA TEMP0
        JSR UPSCORE
        LDX INDEX1              RESTORE CHARGE INDEX
        ENDIF
        LDA X,CHARCO
        CMP I,2
        IFCS                    CHARGE EXHAUSTED
        LDA I,0                 YES. DEACTIVATE IT
        STA X,CHARY
        DEC CHACOU
        ENDIF
        ENDIF
        RTS
        .PAGE
        .SBTTL PLAY - FIRE PLAYER CHARGE
FIREPC
        LDA CURSL2
        IFPL                    PLAYER ALIVE
        LDA QSTATUS
        IFPL                    ATTRACT
        LDA CURMOD              YES. AUTO FIRE
        STA TEMP0
        LDX I,NICHARG+NINVAD-1
        BEGIN                   LOOP FOR EACH INVADER   SHOT UNTIL EXHAUSTED OR CL
        LDA X,CHARY+NPCHAR
        IFNE                    ACTIVE
        LDA X,CHARL1+NPCHAR     YES CALUCLATE ABSOLUTE VALUE OF LINE DELTA
        SEC
        SBC CURSL1
        IFMI
        EOR I,0FF
        CLC
        ADC I,1
        ENDIF
        CMP I,2
```

```
        IFCC                    TOO CLOSE
        INC TEMPO               YES. FIRE
        ENDIF
        ENDIF
        DEX
        MIEND
        LDA TEMPO
        ELSE
        LDA SWSTAT
        AND I,MFIRE
        ENDIF
        IFNE                    FIRE CHARGE
        LDX I,NPCHARG-1         YES
        BEGIN                   LOOP UNTIL VACANCY IS FOUND
        LDA X,CHARY
        IFEQ                    VACANCY
                                YES FIRE CHARGE
        INC CHACOU
        LDA CURSY               START AT CURSOR
        STA X,CHARY
        LDA CURSL1
        STA X,CHARL1            STARTS AT SAME LINE AS CURSOR
        LDA CURSL2
        STA X,CHARL2
        LDA I,0                 0 COLLISION COUNTER
        STA X,CHARCO
        JSR SLAUNC              LAUNCH SOUND
        LDA CURSY
        JSR COLCHK              CHECK FOR COLLISION
        LDX I,0                 EXIT LOOP
        ENDIF
        DEX
        MIEND
        ENDIF
        ENDIF
        RTS
        .PAGE
        .SBTTL PLAY - FIRE INVADER CHARGE
FIREIC
        LDA CURSL2
        IFPL                    PLAYER ALIVE
        LDX I,NINVAD-1          YES.
        BEGIN                   LOOP FOR EACH INVADER
        LDA X,INVAY
        IFNE                    ACTIVE
        CMP I,ILINLIY+20        YES
        IFCS                    INVADER LOW ENOUGH
        LDA X,INVAC2            YES
        AND I,INVFIR
        IFNE                    INVADER MOVING BOTH LEGS ON LINES
        DEC X,INVACT            YES. UPDATE INVADER S FIRE TIMR
        IFMI
        INC X,INVACT
        LDA X,INVAC1
        AND I,INVMOT
        IFEQ
        LDA RANDOM
        LDY ESHCOU
        CMP Y,CHANCE
        IFCS                    TIMER IN FIRE WINDOW
```

```
         LDY WCHAMX
         BEGIN                      LOOP THRU EACH INVADER CHARGE
         LDA Y,CHARY+NPCHARG        UNTIL VACANCY
         IFEQ                       VACANCY
         LDA X,INVAY                YES
         STA Y,CHARY+NPCHARG        START AT INVADER LOC
         LDA X,INVAL1
         STA Y,CHARL1+NPCHARG       SAME LINE AS INVADER
         LDA X,INVAL2
         STA Y,CHARL2+NPCHARG
         LDA WCHARFR
         STA X,INVACT               RESTART TIMER
         JSR ESLSON
         INC ESHCOU
         LDY I,0                    EXIT LOOP
         ENDIF
         DEY
         MIEND
         ENDIF
         ENDIF
         ENDIF
         ENDIF
         ENDIF
         ENDIF
         DEX
         MIEND
         ENDIF
         RTS
CHANCE   .BYTE 0,0E0,0F0,0FA,0FF  HIGHER CHANCE FOR ENEMY SHOT IF LESS ON SCREEN
         .PAGE
         .SBTTL  PLAY-START EXPLOSION
 OUTPUT X AND Y PRESERVED
         SAVEX,SAVEY,TEMP0,1,2,3,4 ARE GARBAGE

         .SBTTL  PLAY-EXPLOSION OF FUSE INIT

INCFS2   STX INDEX1
         LDA I,0FF                  MARK SHOT USED
         STA X,CHARCO
         TYA               CONVERT SHOT INDEX TO INVADER INDEX
         SEC
         SBC I,NICHAR
         TAY
         LDA Y,INVAL1
         STA TEMP4
         LDA RAND02
         AND I,7
         CMP I,3
         IFCS                       RANDOMLY CHOOSE 0 250 ,1 500 , OR 2 750
         LDA I,0
         ENDIF
         PHA
         CLC
         ADC I,CFTYPE
         JSR GEXIFU                 INITIALIZE EXPLOSION
         JSR KILINV                 KILL FUSE
         PLA
         CLC
         ADC I,5
         TAX
```

```
          JSR UPSCOR                UPDATE SCORE
          LDX INDEX1
          RTS
INIPSQ    LDA I,IPTYPE
          JSR DEADCU                KILL CURSOR
          DEC CURSL2                DISPLAY CURSOR
          RTS
INFPSQ    LDA I,FPSPXI              SPECIAL BANG PIC CODE
          BNE INCP2
INPPSQ    LDA I,PPSPXI              SPECIAL EXPLOSION PIC CODE
          BNE INCP2
INCPSQ    LDA I,CPSPXI              SPECIAL EXPLOSION PIC CODE
INCP2     STA SPXIND

          LDA I,CPTYPE
DEADCU                    KILL CURSOR
          STA TEMP3                 EXPOLSION CODE
          LDA CURSY                 POSITION
          STA TEMP0
          LDA CURSL1
          STA TEMP4
          JSR CPEXPL                START NOISE
          JSR GENEX2                INIT EXPLOSION
          LDA I,81                  KILL CURSOR/NO DISP
          STA CURSL2
          LDA I,1                   INIT TIMER FOR EXP.
          STA SPFTIM
          RTS
INCCSQ    JSR CCEXPL                CHARGE-CHARGE
          LDA Y,CHARY+NPCHAR
          STA TEMP0
          LDA Y,CHARL1+NPCHAR
          STA TEMP4
          LDA I,CCTYPE
          JSR GENEXP
          LDA I,0                   DEACTIVATE SHOT
          STA Y,CHARY+NPCHAR
          DEC ESHCOU                ONE LESS SHOT
          LDA I,0FF                 SHOT USED FLAG
          STA X,CHARCO
          RTS
INCIS2
          LDA I,0FF                 SHOT USED MARKER
          STA X,CHARCO
          TYA             CONVERT SHOT INDEX TO INVADER INDEX
          SEC
          SBC I,NICHAR
          TAY
INCISQ    LDA Y,INVAC1
          AND I,ZROCCW ZMOTJM
          CMP I,ZROCCW ZMOTJM
          IFNE                                FLIPPING CCW
          LDA Y,INVAL1                        NO. USE BASE LEG
          ELSE
          LDA Y,INVAL1                        YES. ADJUST BASE LIVE
          SEC
          SBC I,1
          AND I,0F
          ENDIF
          STA TEMP4
```

```
        LDA I,CITYPE
        JSR GEXIFU              INITIALIZE BANG PIC
        JSR KILINV              KILL INVADER
        LDA Y,INVAC1
        AND I,INVABI
        TAY
        LDX Y,INVPIN            INDEX FOR PTS TO ADD
        JMP UPSCOR              UPDATE SCORE
INVPIN  .BYTE 1,2,3,4,1
GEXIFU  PHA
        JSR CIEXPL              BANG SOUND
        LDA Y,INVAY
        STA TEMP0
        PLA

                                GENERAL EXPLOSION STARTER
                                INPUT ACC EXPLOSION TYPE
GENEXP                                  TEMP0 EXPLOSION Y  TEMP4 EXPLOSION LINE
        STA TEMP3               SAVE TYPE   DEPTH
GENEX2  STX SAVEX
        STY SAVEY
        LDA I,0
        STA TEMP1
        STA TEMP2
        LDX I,NEXPLO-1
        BEGIN                   LOOP UNTIL VACANCY IS FOUND
        LDA X,EXPLOY
        BEQ GOTEXP              EXIT IF VACANCY
        LDA X,EXPLOS
        CMP TEMP1
        IFCS                    FURTHEST ALONG SO FAR
        STA TEMP1               YES. SAVE IT
        STX TEMP2
        ENDIF
        DEX
        MIEND
        DEC EXPCOU              WILL BE INCD LATER
        LDX TEMP2               NO VACANCIES. USE FURTHEST AONG
GOTEXP  LDA I,0
        STA X,EXPLOS            START SEQUENCES
        LDA TEMP3
        STA X,EXPLOT            EXPLOSION TYPE
        LDA TEMP0
        STA X,EXPLOY            EXPLOSION DEPTH
        LDA TEMP4
        STA X,EXPLOL            EXPLOSION LINE
        INC EXPCOU              INC COUNTER
        LDX SAVEX
        LDY SAVEY
        RTS
IPTYPE 5                        EXPLOSION TYPE CODES
CPTYPE 1
CCTYPE 0
CITYPE 0
CFTYPE 2
        .PAGE
        .SBTTL  PLAY-PROCESS EXPLOSIONS
PROEXP
        LDA EXPCOU
        IFNE                    ANY BANGS
```

```
        LDA I,0                     YES CLEAR COUNT
        STA EXPCOU
        LDX I,NEXPLO-1
        BEGIN                       LOOP FOR ACH EXPLOSION
        LDA X,EXPLOY
        IFNE                        ACTIVE BANG
        LDA X,EXPLOS                YES. UPDATE SEQUENCES
        LDY X,EXPLOT
        CLC
        ADC Y,TEXINC
        STA X,EXPLOS
        CMP Y,TEXPDN
        IFCS                        EXPLOSION DONE
        LDA I,0                     YES. DEACTIVATE IT
        STA X,EXPLOY
        ELSE
        INC EXPCOU                  NO. INC COUNTER
        ENDIF
        ENDIF
        DEX
        MIEND
        ENDIF
        RTS
TEXPDN  .BYTE 10,15,20,20,20,10          LAST SEQUENCE # TABLE *4
TEXINC  .BYTE 3,1,3,3,3,3
        .PAGE
        .SBTTL PLAY - COLLISION  MAINLINE
COLLIS
        LDX I,NPCHAR-1
        BEGIN                       LOOP FOR EACH PLAYER CHARGE
        LDA X,CHARY
        IFNE                        PLAYER CHARGE ACTIVE
        JSR COLCHK
        ENDIF
        DEX
        MIEND                       ENDLOOP FOR PLAYER CHARGES
        RTS
CHKSM4          .BYTE QCHKS4
        .PAGE
        .SBTTL  PLAY - COLLISION - SINGLE CHECK

 INPUT ACC PLAYER CHARGE Y

COLCHK
        STA TEMPX
        LDY I,NICHAR-1+NINVAD       YES.
        BEGIN                       LOOP FOR EACH INVADER CHARGE    INVADER
        LDA Y,CHARY+NPCHAR
        IFNE                        I C OR INVADER ACTIVE
        CMP TEMPX                   YES. DETERMINE OBSOLUTE DELTA
        IFCS
        SBC TEMPX
        ELSE
        LDA TEMPX
        SEC
        SBC Y,CHARY+NPCHAR
        ENDIF
        CPY I,NICHAR
        IFCC                        ENEMY SHOT OR INVADER
        CMP CHACHA                  SHOT
```

```
        IFCC                    IN RANGE
        LDA Y,CHARL1+NPCHAR     YES.
        EOR X,CHARL1
        IFEQ                    ON SAME LINE
        JSR INCCSQ              YES. INITIALIZE EXPLOSION
        ENDIF
        ENDIF
        ENDIF
        ELSE
        PHA                     INVADER. SAVE DELTA
        STY INDEX2
        LDA Y,INVAC1-NICHAR
        AND I,INVABI
        TAY
        PLA
        CMP Y,ENSIZE
        IFCC                    IN RANGE BY TYPE
        CPY I,ZABFUS            YES.
        IFEQ                    FUSE
        LDY INDEX2              YES.
        LDA Y,INVAY-NICHAR
        CMP CURSY
        IFNE                    FUSE AT TOP
        LDA X,CHARL1            NO.
        CMP Y,INVAL1-NICHAR
        IFEQ                    SAME BASE LINE
        LDA Y,INVAL2-NICHAR     YES.
        IFMI                    VULNERABLE FUSE
        JSR INCFS2              YES. START BANG, KILL FUSE, GIVE PTS.
        ENDIF
        ENDIF
        ENDIF
        ELSE
        LDY INDEX2              NO. FLIPPER,TANKER,SPINNER,PULSAR
        LDA Y,INVAL2-NICHAR
        IFMI                    FLIPPER
        LDA Y,INVAL1-NICHAR     YES.
        CMP X,CHARL2            BASE   SECONDARY MATCH
        BEQ YESCOL
        BNE OKATOP              NO. CHECK FOR BASE MATCH
        ENDIF
        LDA Y,INVAY-NICHAR
        CMP CURSY
        IFNE                    AT TOP
OKATOP  LDA Y,INVAL1-NICHAR     NO.
        CMP X,CHARL1
        IFEQ                    BASE LEG MATCH
YESCOL  STX INDEX1              YES.
        JSR INCIS2              START BANG
        LDX INDEX1
        ENDIF
        ENDIF
        ENDIF
NOCOL
        ENDIF
        LDY INDEX2
        ENDIF
        ENDIF
        DEY
        MIEND                   ENDLOOP FOR ICS
        LDA X,CHARCO
```

```
        CMP I,0FF
        IFEQ                    PLAYER CHARGE SPENT
        LDA I,0                 YES. DEACTIVATE IT
        STA X,CHARY
        DEC CHACOU
        STA X,CHARCO
        ENDIF
        RTS

        .PAGE
        .SBTTL PLAY - ANALYZE GAME
ANALYZ
        LDA CURSL2
        IFMI                    CURSOR DEAD
        LDA CHACOU              YES
        ORA ESHCOU
        ORA EXPCOU
        IFEQ                    ANY ACTIVE CHARGES OR BANGS
        LDX WINVMX              NO. DROP EVERYBODY INTO WELL
        BEGIN                   LOOP FOR EACH INVADER
        LDA X,INVAY
        IFNE                    ACTIVE INVADER
        CLC                     YES MOVE IT DOWN
        ADC I,15.
        IFCC
        CMP I,ILINDDY
        ENDIF
        IFCS                    INVADER AT BOTTOM
        LDA I,0                 YES. DEACTIVATE IT
        ENDIF
        STA X,INVAY
        ENDIF
        DEX
        MIEND
        LDX PLAYUP
        LDA X,LIVES1
        CMP I,1
        IFEQ                    GAME OVER
        LDA I,0                 YES. REQUEST RECALC OF WELL TOP
        STA LEVELY
        LDA I,1                 REQUEST REDISPLAY OF WELL
        STA ROTDIS
        LDA EYL
        SEC
        SBC I,20
        STA EYL                 SHRINK HOLE
        LDA EYH
        SBC I,0
        STA EYH
        CMP I,0FA
        CLC
        IFEQ                    FAR ENOUGH
        SEC                     YES. END GAME
        ENDIF
        ELSE
        LDA CURSY               MOVE CURSOR DOWN
        CLC
        ADC I,15.
        STA CURSY
        IFCC
```

```
        CMP I,ILINDDY
        ENDIF
        ENDIF
        IFCS                    CURSOR AT BOTTOM
                                YES. END OF LIFE PHASE.
        LDA I,CENDLI            YES. GO TO END OF LIFE STATE
        STA QSTATE
        JSR INICHA              CLEAR CHARGES
        LDA INMCOU              ADD # OF INVADERS
        CLC
        ADC INCCOU
        CLC
        ADC NYMCOU              TO # NYMPHS
        CMP I,NNYMPH-1
        IFCS                    MAX OUT
        LDA I,NNYMPH-1
        ENDIF
        STA NYMCOU              FOR NEXT LIFE
        ENDIF
        ENDIF
        ELSE
ZQVAVG          LDA QT3
        ORA QT6
        IFNE
        LDA I,17
        CMP LSCORH
        IFCC
        LDX LSCORL
        INC X,0
        ENDIF
        ENDIF
        LDA CURMOD
        IFEQ                    TOP MODE
        LDA NYMCOU              YES CURSOR ALIVE   BANGS DONE
        ORA EXPCOU
        IFEQ                    ALL NYMPHS CONVERTED
        LDY WINVMX              YES. ALL INVADERS OOF LINES
        BEGIN                   LOOP FOR EACH INVADER UNTIL ALL CHECKED ON LINE FO
        LDA Y,INVAY
        IFNE
        CMP I,11
        BCS LINER               EXIT IF LINER  NOT AT TOP
        ENDIF
        DEY
        MIEND                   EXIT AFTER ALL CHECKED.  NO LINERS
        JSR INDROP              YES.
        JSR INICHA              CLEAR CHARGES
        ENDIF
LINER   LDA SWSTRT
        AND I,MSTRT2 MSTRT1
        IFNE                    EITHER START PRESSED
        BIT QSTATUS             YES
        IFMI                    ATTRACT
        LDA OPTIN1              NO.
        AND I,43
        CMP I,40
        IFEQ                    FREE PLAY   ABORT ENABLED
        JSR INDROP              YES. INITIATE DROP MODE
        ENDIF
        ENDIF
```

```
        ENDIF
        ENDIF
        ENDIF
        RTS
        .PAGE
        .SBTTL  INITIALIZE CURSOR DROP MODE

INDROP
        LDA I,CDROP             DROP STATE NEXT
        STA QSTATE
        LDA CURMOD              SET CURSOR DROP MODE
        ORA I,80
        STA CURMOD
        LDA I,0                         INITIALIZE DOWNWARD ACCELERATION
        STA CURSVL
        STA CURSYL              ZERO FRAC. POSITION
        STA EYLL                TO PREVENT JERKING
        STA ELICNT
        LDA I,2
        STA CURSVH
        LDX I,NLINES-1
        BEGIN
        LDA X,LINEY
        IFNE
        INC ELICNT              COUNT LIVE SPIKES
        ENDIF
        DEX
        MIEND
        LDA ELICNT
        IFNE                    ENEMY LINES
        LDA CURWAV              YES.
        CMP I,7
        IFCC                    WARN PLAYER
                                YES
        LDA I,6*QUASEC          WARNING DELAY
        STA QTMPAUS
        LDA I,CPAUSE            PAUSE FIRST
        STA QSTATE
        LDA I,CDROP             THEN DROP MODE
        STA QNXTSTA
        LDA I,80                SET WARNING FLAG
        STA ELICNT
        ENDIF
        ENDIF
        LDA I,-1
        STA SUZTIM              DEACTIVATE SUPERZAPPER
        RTS
        .PAGE

        .SBTTL  PLAY-PROCESS BIG BOOM

PRBOOM  LDA BOOMTI                      SET BOOM OFF FLAG
        STA BOOMFL
        LDX I,NPARTI-1
        STX INDEX1
        BEGIN                   LOOP FOR EACH PARTICLE
        LDX INDEX1
        LDA X,PARTIY
        IFEQ                    ACTIVE PARTICLE
        LDA BOOMTI              NO.
```

```
        IFNE                    BOOM TIMER EXPIRED
        JSR TIMLAU              NO. LAUNCH MORE PARTICLES OF TIME
        ENDIF
        ELSE
        JSR UPARPO              YES. UPDATE PARTICLE POSITION
        JSR DECPAR              DECELERATE PARTICLE
        LDA I,-1                BOOM ACTIVE
        STA BOOMFL
        ENDIF
        DEC INDEX1
        MIEND                   END LOOP
        LDA QFRAME
        AND I,1
        IFEQ
        LDA BOOMTI
        IFNE
        DEC BOOMTI              UPDATE BOOM TIMER  STOP AT 0
        ENDIF
        ENDIF
        LDA BOOMFL
        IFEQ                    BOOM ACTIVE
        LDA I,CGETIN            NO. GET INITIALS
        STA QSTATE
        ENDIF
        RTS
        .PAGE
TIMLAU
        LDA QFRAME
        AND I,0
        IFEQ                    DELAY SINCE LAST LAUNCH OK
                                YES. LAUNCH ANOTHER
        LDA I,80                SET UP INITIAL LOCATION IN CENTER
        STA X,PARTIX
        STA X,PARTIY
        STA X,PARTIZ
                                SET UP VELOCITY  RANDOM WITHIN
        LDA RANDO2              GIVE RANGE
        STA X,PARLXV            FRACTIONAL X VELOCITY
        JSR FIXTOP
        STA X,PARTXV            INTEGER X
        LDA RANDOM
        STA X,PARLYV            Y
        JSR FIXTOP
        IFPL                    UPDATE PARTICLE POSITION
        EOR I,0FF
        CLC
        ADC I,1
        ENDIF
        STA X,PARTYV
        LDA RANDOM              Z
        STA X,PARLZV
        JSR FIXTOP
        STA X,PARTZV
        JSR CIEXPL              MAKE NOISE
        ENDIF
        RTS
FIXTOP
        LSR
        LDA RANDO2
        AND I,7
```

```
        IFCS
        EOR I,0FF
        CLC
        ADC I,1
        ENDIF
        RTS
        .PAGE
UPARPO                          UPDATE PARTICLE POSITION
        LDA X,PARLYV            Y
        CLC
        ADC X,PARLIY
        STA X,PARLIY            FRACTIONAL
        LDA X,PARTYV
        IFPL
        ADC X,PARTIY            + VELOCITY
        CMP I,0F0
        IFCS
        LDA I,0                 OFF SCREEN
        ENDIF
        ELSE
        ADC X,PARTIY            - VELOCITY
        CMP I,10
        IFCC
        LDA I,0                 OFF SCREEN
        ENDIF
        ENDIF
        TAY
        LDA X,PARLXV            X
        CLC
        ADC X,PARLIX
        STA X,PARLIX            FRACTIONAL
        LDA X,PARTXV
        IFPL
        ADC X,PARTIX            +VELOCITY
        CMP I,0F0
        IFCS
        LDY I,0                 OFF SCREEN
        ENDIF
        ELSE
        ADC X,PARTIX            -VELOCITY
        CMP I,10
        IFCC
        LDY I,0                 OFF SCREEN
        ENDIF
        ENDIF
        STA X,PARTIX
        LDA X,PARLZV            Z
        CLC
        ADC X,PARLIZ
        STA X,PARLIZ            FRACTIONAL
        LDA X,PARTZV
        IFPL
        ADC X,PARTIZ            + VELOCITY
        CMP I,0F0
        IFCS
        LDY I,0                 OFF SCREEN
        ENDIF
        ELSE
        ADC X,PARTIZ            VELOCITY
        CMP I,10
```

```
        IFCC
        LDY I,0                 OFF SCREEN
        ENDIF
        ENDIF
        STA X,PARTIZ
        TYA
        STA X,PARTIY
        RTS
        .PAGE
DECPAR
        LDA I,-3                VELOCITY 0 COUNTER
        STA TEMP0
        LDA X,PARLXV
        LDY X,PARTXV
        JSR DECELE              DECELERATE X VELO
        STA X,PARLXV
        TYA
        STA X,PARTXV
        LDA X,PARLYV
        LDY X,PARTYV
        JSR DECELE              DECELERATE Y VELO
        STA X,PARLYV
        TYA
        STA X,PARTYV
        LDA X,PARLZV
        LDY X,PARTZV
        JSR DECELE              DECELERATE Z VELO
        STA X,PARLZV
        TYA
        STA X,PARTZV
        LDA TEMP0
        IFEQ                    ALL 3 DIRECTIONS VELOCITY 0
        STA X,PARTIY            YES. DEACTIVATE PARTICLE
        ENDIF
        RTS
DECELE
        STY TEMP2
        BIT TEMP2
        IFPL                    VELOCITY+ OR -
        SEC                     + SO ECELERATE BY SUBTRACTING
        SBC DECELO
        STA TEMP1
        LDA TEMP2
        SBC I,0
        BCC HIT0                VELOCITY HIT 0   BR IF YES
        ELSE
        CLC                     -, SO DECELERATE BY ADDING
        ADC DECELO
        STA TEMP1
        LDA TEMP2
        ADC I,0
        IFCS                    VELOCITY HIT 0
HIT0    INC TEMP0               YES INCREMENT VELOCITY 0 COUNTER
        LDA I,0
        STA TEMP1
        ENDIF
        ENDIF
        TAY                     RETURN WITH NEW VELOCITY
        LDA TEMP1
        RTS
```

```
DECELO  .BYTE 20
        .PAGE
        .SBTTL  INITIALIZE PARTICLES
INBOOM
        LDX I,NPARTI-1
        BEGIN
        LDA I,0
        STA X,PARTIY                DEACTIVATE PARTICLE
        DEX
        MIEND
        LDA I,020                   1/5 SECOND UNTIS
        STA BOOMTI
        STA BOOMFL                  ACTIVATE BOOM
        LDA I,CDBOOM                BOOM DISPLAY STATE
        STA QDSTATE
        LDA I,0
        STA ZADJL
        STA ZADJL+1
        RTS
        .SBTTL UTILITY - LINE LINE POLOR DELTA
 INPUT  Y,ACC LINE # FOR DETERMINATIN

 OUTPUT ACC # OF LINES ACC LINE IS FROM Y LINE IN
        SHORTEST DIRECTION  -8 TO +7   -MEANS CLOCKWISE

POLDEL
        STY TEMP1
        SEC
        SBC TEMP1
        STA TEMP1
        BIT WELTYP
        IFPL                        PLANAR
        AND I,0F
        BIT A,EIGHT                 NO.
        IFNE                        TAKE SHORTEST ROUTE
        ORA I,0F8
        ENDIF
        ENDIF
        RTS
EIGHT   .BYTE 8

        .SBTTL  INITIALIZE-PLANES OF STARS

INSTAR
        LDX I,NPLANE-1
        LDA I,0
        BEGIN                       DEACTIVATE ALL PLANES
        STA X,PLANEY
        DEX
        MIEND
        LDA I,0F0
        STA PLANEY+NPLANE-1         ACTIVATE LAST PLANE FAR AWAY
        LDA I,0FF
        STA PLAGRO                  SET STAR FIELD GROWING FLAG
        RTS
        .PAGE
        .SBTTL  PLAY-PROCESS PLANES OF STARS
 INPUT  IF PLAGRO IS-,THEN STAR FIELD IS STILL GROWING
        IF PLAGRO IS 0,THEN STAR FIELD IS DEACTIVATED
 OUTPUT IF PLAGRO IS 0,THEN STAR FIELD IS COMPLETELY DEAD
```

```
PRSTAR  LDA PLAGRO              STAR FIELD ACTIVE
        IFNE
        LDA I,0                 YES. PROCESS PLANES
        STA TEMPO               CLEAR COUNT OF ACTIVE PLANES
        LDX I,NPLANE-1
        STX INDEX1
        BEGIN                   LOOP FOR EACH PLANE
        LDX INDEX1
        LDA X,PLANEY
        IFNE                    PLANE ACTIVE
        SEC                     YES.
        SBC I,07                UPDATE PLANE POSITION
        IFCS
        CMP I,10
        ENDIF
        IFCC                    TOO CLOSE
        LDY PLAGRO              YES
        IFMI                    STILL GROWING
        LDA I,0F0               YES. START AT FARTHEST POINT
        ELSE
        LDA I,0                 NO. DEACTIVATE
        ENDIF
        ENDIF
        ELSE
        LDY PLAGRO              NO. STILL GROWING
        IFMI
        TXA                     YES.
        CLC
        ADC I,1                 GET INDEX OF PREVIOUS PLANE
        CMP I,NPLANE
        IFCS
        LDA I,0
        ENDIF
        TAY
        LDA Y,PLANEY            PREVIOUS PLANE ACTIVE
        IFNE
        CMP I,0D5               YES.
        IFCC                    IS PREVIOUS PLANE CLOSE ENOUGH
        LDA I,0F0               YES. START NEW PLANE
        ELSE
        LDA I,0                 NO. STILL INACTIVE
        ENDIF
        ENDIF
        ENDIF
        ENDIF
        STA X,PLANEY
        ORA TEMPO
        STA TEMPO
        DEC INDEX1
        MIEND
        LDA TEMPO
        IFEQ
        STA PLAGRO
        ENDIF
        ENDIF
        RTS
        .PAGE

        .SBTTL INITIALIZE SUPER ZAP
CSUMAX 2
```

```
CSUINT 1
CSUSTA 3
INISUZ
        LDA I,0                 SET SUPZAP USE COUNTER AND TIMER TO 0.
        STA SUZCNT
        STA SUZTIM
        RTS
        .SBTTL PROCESS SUPER ZAPPER
PROSUZ
        LDA QSTATUS
        IFMI                    ATTRACT
        LDA SUZTIM              NO
        IFEQ                    ZAP ACTIVE
        LDA CURSL2              NO.
        IFPL                    CURSOR ALIVE
        LDA SWFINA              YES
        AND I,MSUZA
        IFNE                    ZAP PRESSED
        LDA SUZCNT              YES.
        CMP I,CSUMAX
        IFCC                    ZAPS LEFT
        INC SUZCNT              YES. UPDATE ZAP COUNTER
        LDA I,1
        STA SUZTIM              START ZAP TIMER

        ENDIF
        LDA SWFINA
        AND I, C MSUZA MFAKE
        STA SWFINA
        ENDIF
        ENDIF
        ELSE
        INC SUZTIM              YES. ZAP ACTIVE
        LDX SUZCNT
        LDA SUZTIM
        CMP X,TIMAX
        IFCS                    ZAP TIMER EXPIRED
        LDA I,0
        STA SUZTIM              YES. DEACTIVATE ZAP
        ENDIF
        JSR KILENE              WIPE OUT INVADERS   CHARGES
        ENDIF
        ENDIF
        LDA SWFINA
        AND I, CMFAKE
        STA SWFINA              CLEAR  SWITCH NOT PROCESSED  FLAG
        RTS

TIMAX   .BYTE 0,CSUSTA+ 8* CSUINT+1  ,CSUSTA+ 1* CSUINT+1  ,0,0

        .SBTTL SUPER ZAP-WIPE OUT ENEMY
KILENE  LDA SUZTIM
        CMP I,CSUSTA
        IFCS
        AND I,CSUINT
        IFEQ                    TIME FOR ANOTHER WIPE OUT

        LDY WINVMX              YES.

        BEGIN                   LOOP THRU INVADERS
```

```
        LDA Y,INVAY
        BNE EXIKIL                  SPECIAL EXIT FOR 1ST LIVE ONE
        DEY                         EXIT LOOP IF ALL ARE DEACTIVE
        MIEND
        LDA I,0                     ALL ARE DEAD. DEACTIVATE ZAP
        STA SUZTIM
        ENDIF
        ENDIF
        RTS

EXIKIL
        LDA Y,INVAC2                MAKE SURE IT S NOT A CARRIER
        AND I, C INVCAR
        STA Y,INVAC2
        JMP INCISQ                  START EXPLOSION
CHKSM5          .BYTE QCHKS5
        HLL65
        .END
```