

TITL 'H O P P E R'  
OPTION XREF

\*\*\*\*\*  
\* DESIGNED BY MICHAEL ARCHULETA \*  
\* PROGRAMED BY JOHN PHILLIPS AND \*  
\* MICHAEL ARCHULETA \*  
\* DATE: FEB 4, 1983 \*  
\* \*  
\* H O P P E R \*  
\*\*\*\*\*

RORG >6000  
BYTE >EE  
BYTE 0  
DATA HOPPER  
BYTE TEND-#-1  
TEXT 'HOPPER'

4

TEND EQU #

\*\*\*\*\*  
\* C P U R A M E Q U A T E S \*  
\*\*\*\*\*

CPURAM EQU >8300  
MYWS EQU CPURAM  
MYWS2 EQU CPURAM+>A0 FOR BLWP'S  
CHSVDR EQU CPURAM+>20 CHASERS SAVED DIR TABLE  
CHSODR EQU CHSVDR+0  
CHS1DR EQU CHSVDR+1  
CHS2DR EQU CHSVDR+2  
CHATIM EQU CPURAM+>23 CHASERS TIME COUNTER TABLE  
CHOTIM EQU CHATIM+0  
CH1TIM EQU CHATIM+1  
CH2TIM EQU CHATIM+2  
GAMLV1 EQU CPURAM+>26 PLAYER 1 GAME LEVEL  
GAMLV2 EQU CPURAM+>27 PLAYER 2 GAME LEVEL  
SAVDLY EQU CPURAM+>28 SAVED CHASER DELAY COUNTER  
FLAG EQU CPURAM+>2A FOR DEMO MODE  
RANCNT EQU CPURAM+>30 COUNTER FOR ENTRAPMENT CHECKP  
RANINC EQU CPURAM+>31 POINTS TO NEXT DIRECTION TO TRY  
DEMDLY EQU CPURAM+>32 FOR DEMO DELAY  
HOWMNY EQU CPURAM+>34 HOW MANY PLAYERS 0=1, 1=2  
WHICH1 EQU CPURAM+>35 WHICH PLAYER 1 OR 2  
SCADDR EQU CPURAM+>36 2 BYTES, SCORE ADD LOCATION  
KANGS1 EQU CPURAM+>38 PLAYER 1'S KANGOS  
KANGS2 EQU CPURAM+>39 PLAYER 2'S KANGOS  
PAUSEF EQU CPURAM+>40 PAUSE FLAG 0-NO PAUSE MSG 1-PAUSE MSG  
PAUSET EQU CPURAM+>42 PAUSE MSG TIMER  
KPOS EQU CPURAM+>82  
KCHAR EQU CPURAM+>84  
CHAPOS EQU CPURAM+>86 CHASER POSITION  
CHAPS1 EQU CHAPOS+>00 CHASER 1  
CHAPS2 EQU CHAPOS+>02 CHASER 2  
CHAPS3 EQU CHAPOS+>04 CHASER 3  
CHAKIL EQU CPURAM+>8D CHASERS KILLED  
KEYDLY EQU CPURAM+>8E KEYSKAN DELAY LOOP  
BONUS EQU CPURAM+>90 ALL THE FOLLOWING  
DIGTHB EQU CPURAM+>92 ARE FOR SCORE  
DIGTLB EQU CPURAM+>93 "  
CARYHB EQU CPURAM+>94 "  
CARYLB EQU CPURAM+>95 "  
SAVNUM EQU CPURAM+>96 "  
SAVSAD EQU CPURAM+>98 "  
DEM EQU CPURAM+>9A FOR DEMO MODE  
NWAY EQU CPURAM+>9C "

```

CNTR EQU CPURAM+>9E "
* R E S E R V E D   E Q U A T E S
KEYBRD EQU CPURAM+>74
KEY EQU CPURAM+>75
JOYY EQU CPURAM+>76
JOYX EQU CPURAM+>77
RANDOM EQU CPURAM+>78
TIMER EQU CPURAM+>79
MOTION EQU CPURAM+>7C
STATUS EQU CPURAM+>7C
INTWS EQU CPURAM+>C0
SEED EQU CPURAM+>C0 BUILT-IN SEED
DISINT EQU CPURAM+>C2 DISABLE INTERRUPT REG
GPLWS EQU CPURAM+>E0
SVVDP1 EQU CPURAM+>D4
SCNTIM EQU CPURAM+>D6

```

```

*****
* MEMORY MAPPED EQUATES *
*****

```

```

SIT EQU >0000
SAL EQU >0300
CT EQU >0380
LOWVDP EQU >03A0 FREE SPACE(>60 BYTES)
SDT EQU >0400
SVT EQU >0780
PDT EQU >0800
VDPWA EQU >8C02
VDPWD EQU >8C00
VDPRD EQU >8800
SCAN EQU >000E

```

```

*****
* SOUND LIST EQUATES *
*****

```

```

SNDBUF EQU >1000
HOPSND EQU SNDBUF HOP KANGO SOUND
SLDSND EQU HOPSND+29 SLIDE CRATE SOUND
DILSND EQU SLDSND+49 DIE LEFT SOUND
DIRSND EQU DILSND+5 DIE RIGHT SOUND
APRSND EQU DIRSND+5 APPEARING KANGAROO
MLTSND EQU APRSND+14 MELTING SOUND
SNDOFF EQU MLTSND+49 SOUNDS OFF
* EQU SNDOFF+6 OTHER SOUND

```

```

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

```

```

VDPADD EQU 0
WCOUNT EQU 2
RCOUNT EQU 2
WL0C EQU 3
RL0C EQU 3
VADDLB EQU MYWS+1
ROLB EQU MYWS+1
R1LB EQU MYWS+3
R2LB EQU MYWS+5
R3LB EQU MYWS+7
R4LB EQU MYWS+9
R5LB EQU MYWS+11
R6LB EQU MYWS+13
R7LB EQU MYWS+15
R8LB EQU MYWS+17
R9LB EQU MYWS+19
R10LB EQU MYWS+21

```

R11LB EQU MYWS+23  
R12LB EQU MYWS+25  
R13LB EQU MYWS+27  
R14LB EQU MYWS+29  
R15LB EQU MYWS+31  
R16LB EQU MYWS+33

\*\*\*\*\*  
\* DATA AND BYTE STATEMENTS NEEDED  
\*\*\*\*\*

H00 BYTE >00  
H01 BYTE >01  
H02 BYTE >02  
H03 BYTE >03  
H04 BYTE >04  
H05 BYTE >05  
H06 BYTE >06  
H07 BYTE >07  
H08 BYTE >08  
H09 BYTE >09  
H0A BYTE >0A  
H0B BYTE >0B  
H0C BYTE >0C  
H0D BYTE >0D  
H0F BYTE >0F  
H10 BYTE >10  
H11 BYTE >11  
H12 BYTE >12  
H14 BYTE >14  
H20 BYTE >20  
H39 BYTE >39  
HA1 BYTE >A1  
HCO BYTE >C0  
HE2 BYTE >E2  
HFF BYTE >FF  
BCOLOR BYTE 4, 6, 12, 13, 5, 8, 2, 7, 3, 9 BACKGROUND COLORS  
PAUSFL BYTE 4, 4, 4, 4, 4 FILLER FOR PAUSE MSG  
KDELAY BYTE 10, 10, 10, 10, 10, 10, 10, 9, 8, 7  
H0000 DATA >0000  
ZSCORE DATA >3030, >3030, >3030 6 ZEROES

FOR SCORE ROUTINE  
FOR FLASHING  
OFF SCREEN FOR SPRITES  
VDP R1  
NEGATIVE ONE

\*  
KANINI DATA >6760, >8C09  
CRTINI DATA >C001, >880A  
CHAINI DATA >5710, >8404, >5710, >8006  
DATA >57B0, >840F, >57B0, >800D  
DATA >0758, >8408, >0758, >800C  
DATA >D000

\* == THAT WAS 30 BYTES OF SAL INITS

WHERE DATA DOWN  
DATA BRDINC  
DATA LEFT  
DATA RIGHT  
DATA BRDINC  
DATA UP

WHERTO DATA CUDMOV  
DATA CLRMOV  
DATA CUDMOV  
DATA CLRMOV

RNDJMP DATA UP  
DATA DOWN  
DATA LEFT  
DATA RIGHT  
DATA KICK

PDINT DATA >0000,>0200,>0800,>1600 BONUS SCORES  
 DIRINC DATA >FF00,>FF00,>1000,>1000  
 CHADIR DATA >FC00,>FFFC,>0400,>0004  
 CHADLY DATA 26,24,22,20,18,16,14,12,10,10  
 \* THESE ARE BASED UPON PLAYING LEVEL

\*\*\*\*\*  
 \* TEXT STATEMENTS \*  
 \*  
 \*\*\*\*\*

SCORE1 TEXT 'SCORE1'  
 SCORE2 TEXT 'SCORE2'  
 SCOREM TEXT 'SCORE: '  
 HIGHM TEXT 'HIGH : '  
 TIMEM TEXT 'LEVEL: '  
 PAUSEM TEXT 'PAUSE!' 6 CHARS  
 PRESSM TEXT 'PRESS ANY KEY TO BEGIN' 22 CHRS  
 COPYM TEXT '@1983 TEXAS INSTRUMENTS' 24 CHARS  
 ROUNDM TEXT 'ROUND (0 - 9): ' 22 CHARS  
 ONMSG TEXT 'PRESS: (1) ONE PLAYER' 22 CHARS  
 TWOMSG TEXT ' (2) TWO PLAYER' 22 CHARS  
 GAMED TEXT ' GAME OVER '  
 PRESS1 TEXT ' PRESS: '  
 PRESS2 TEXT ' REDO OR BACK '  
 DONE1 TEXT ' GAME OVER PLAYER 1 ' 20 CHARS  
 READY1 TEXT ' PLAYER 1 GET READY ' 20 CHARS  
 READY2 TEXT ' PLAYER 2 GET READY ' 20 CHARS  
 NEXTPL TEXT 'PRESS A KEY TO BEGIN'  
 NXTPLB TEXT ' '  
 EGO1 TEXT 'PROGRAMMED BY: ' 20  
 EGO2 TEXT ' JOHN PHILLIPS AND '  
 EGO3 TEXT ' MICHAEL ARCHULETA '

\*\*\*\*\*  
 \* CHARACTER DEFINITIONS \*  
 \*  
 \*\*\*\*\*

TCHARS DATA >FCFC,>3C3C,>3C3C,>3F3F UL H  
 DATA >3F3F,>3C3C,>3C3C,>FCFC  
 DATA >3F3F,>3C3C,>3C3C,>FCFC  
 DATA >FCFC,>3C3C,>3C3C,>3F3F  
 \*  
 DATA >0F1F,>3F7F,>FCF8,>F0F0 UL D  
 DATA >F0F0,>F8FC,>7F3F,>1F0F  
 DATA >F0F8,>FCFE,>3F1F,>0F0F  
 DATA >0F0F,>1F3F,>FEFC,>FBFO  
 \*  
 DATA >FFFF,>3F3F,>3C3C,>3C3F UL P  
 DATA >3F3F,>3C3C,>3C3C,>FCFC  
 DATA >F0F8,>FCFE,>1F0F,>1FFF  
 DATA >FEFC,>0000,>0000,>0000  
 \*  
 DATA >FFFF,>3F3F,>3C3C,>3C3F UL R  
 DATA >3F3F,>3C3C,>3C3C,>FCFC  
 DATA >F0F8,>FCFE,>1F0F,>1FFF  
 DATA >FEFC,>6030,>180C,>3F3F  
 \*  
 DATA >FFFF,>3F3F,>3C3C,>3C3F UL P  
 DATA >3F3C,>3C3C,>3F3F,>FFFF  
 DATA >FFFF,>FFFF,>0303,>30F0  
 DATA >F030,>0303,>FFFF,>FFFF  
 \*  
 ACHARS DATA >0000,>000B,>1F7F,>7F7F >80  
 DATA >0020,>60E0,>C0C0,>8000 >81

DATA >C0C0, >E0F0, >F8FC, >7C78 >82  
DATA >183C, >BBBF, >7FFF, >FFFF >83

\*

DATA >3FFE, >FFFF, >FFFF, >FFFF >84  
DATA >0080, >F8FE, >FFFF, >FFFF >85  
DATA >7C7F, >7E7E, >FFFF, >FFFF >86  
DATA >0F1F, >3F7F, >7F7F, >3F1F >87

\*

DATA >F8FC, >FEFE, >FFFF, >FFFE >88  
DATA >1F1F, >0F0F, >0F0F, >0707 >89  
DATA >FEFE, >FEFE, >FCFC, >FCF8 >8A  
DATA >0703, >0303, >0101, >0703 >8B

\*

DATA >FFFF, >FFFF, >FEFC, >E080 >8C  
DATA >FFFF, >F880, >0000, >0000 >8D  
DATA >FFF7, >ED6B, >2301, >0000 >8E  
DATA >FFFF, >FFFF, >FFFF, >7A00 >8F

\*

DATA >F8FB, >F8FB, >F8F0, >C000 >90  
DATA >FFFF, >FFFF, >FFFF, >FFFF >91  
DATA >0000, >0000, >0001, >0307 >92  
DATA >0000, >0000, >0080, >E0F8 >93  
DATA >0000, >0000, >0000, >0000 >94

\*

\* FOR TITLE SCREEN . . . 160 BYTES

\*

ULCORN DATA >FFC0, >A090, >8884, >8281 CHAR 0  
URCORN DATA >FF03, >0509, >1121, >4181 CHAR 1  
LLCORN DATA >8182, >8488, >90A0, >C0FF CHAR 2  
LRCORN DATA >8141, >2111, >0905, >03FF CHAR 3  
HORZCH DATA >FF00, >0000, >0000, >00FF CHAR 4  
VERTCH DATA >8181, >8181, >8181, >8181 CHAR 5  
\* ALL THOSE CHARACTERS GO INTO CHARACTER SET 0  
FILLER DATA >0000, >0000, >0000, >0000 CHAR 8  
CPYRIT DATA >3C42, >99A1, >A199, >423C COPYRIGHT SYMBOL  
\* ALL THOSE CHARS GO INTO CHAR SET 1 YELLOW ON TRANS

\*\*\*\*\*

\* TITLE SCREEN SIT FORMAT STATEMENT \*

\*\*\*\*\*

TITSIT DATA >2020, >2020, >2000, >0220  
DATA >2004, >0620, >2008, >0A20 FIRST LINE OF  
DATA >2008, >0A20, >2010, >1220 TITLE SCREEN  
DATA >200C, >0E20, >2020, >2020 32 BYTES

\*

DATA >2020, >2020, >2001, >0320  
DATA >2005, >0720, >2009, >0B20 SECOND LINE OF  
DATA >2009, >0B20, >2011, >1320 TITLE SCREEN  
DATA >200D, >0F20, >2020, >2020 32 BYTES

\*

TIT AUS DATA >9494, >8081, >8294  
DATA >9283, >8485, >8693 AUSTRALIA  
DATA >8791, >9191, >918B MAP  
DATA >8991, >9191, >918A 30 BYTES  
DATA >8BBC, >8DBE, >8F90 30 BYTES

\*\*\*\*\*

\* SPRITE DEFINITIONS \*

\*

\*\*\*\*\*

FACE DATA >0000, >0000, >0007, >0F19 FACE SPRITE  
DATA >1D1F, >1F1F, >0E0F, >0703 CHARS >80->83  
DATA >0000, >0000, >00F0, >F098  
DATA >8BF8, >F8FB, >70F0, >E0C0

```

HATMOU DATA >1B3C,>7EFF,>7F3C,>1B60 HAT & MOUSTACHE
DATA >8000,>030C,>0000,>0000 CHARS >84->87
DATA >0020,>40B0,>0000,>0000
DATA >0000,>C030,>0000,>0000
CRATES DATA >0000,>0709,>1224,>487F CRATE
DATA >517F,>517F,>5151,>7F00 CHARS >88->8B
DATA >0000,>FE11,>2345,>89F3
DATA >15FB,>15F9,>1214,>FB00
*
LKANG DATA >0000,>0306,>1F0F,>0339
DATA >7D7F,>3D03,>0306,>071F LEFT FACING
DATA >60C0,>C0E0,>E0C0,>80B0 KANGAROO
DATA >C1C1,>E3E6,>FEFC,>78BB CHARS >8C->8F
*
DATA >0603,>0307,>0703,>0101 RIGHT FACING
DATA >8383,>C767,>7F3F,>1E1D KANGAROO
DATA >0000,>C060,>F8F0,>C09C CHARS >90-93
DATA >BEFE,>BCC0,>C060,>E0FB
*
DATA >0202,>0103,>0301,>0123 UP FACING
DATA >2323,>3333,>1F1F,>0704 KANGAROO
DATA >4040,>80C0,>C09B,>DBF0 CHARS >94->97
DATA >C0C0,>C0C0,>E0F0,>E0FB
*
DATA >0202,>0102,>0302,>0301 DOWN FACING
DATA >0705,>0303,>0707,>030E KANGAROO
DATA >4040,>8040,>C040,>80C0 CHARS >98->9C
DATA >E8AB,>C8DB,>F8E0,>C070
*

```

\*\*\*\*\*

```

* CRATE POSITIONS *
* ** EACH SCREEN WILL HAVE 32 CRATES *
*****

```

```

MAZE1 DATA >0209,>0405,>0409,>0415 THRU 06
DATA >0605,>0609,>060D,>060F,>0611,>0613 THRU 12
DATA >0615,>0805,>080D,>0811,>0A05 THRU 18
DATA >0A09,>0A0B,>0A0D,>0A15,>1617 THRU 24
DATA >0C15,>0E05,>0E07,>0E09 THRU 30
DATA >0E11,>0E13,>0E15,>100D,>1015 THRU 36
DATA >1205,>1209,>120B,>120D,>120F,>1211 THRU 42
DATA >1405,>1409,>1411,>1215,>040F THRU 48

```

\*\*\* THERE ARE 40 CRATES TO THE SCREEN

```

MAZE2 DATA >0203,>0207,>0211,>0217 THRU 06
DATA >040D,>0605,>0607,>060D,>060F,>0611 THRU 12
DATA >0613,>0807,>080D,>0813,>0815 THRU 18
DATA >0A07,>0A09,>0C09,>0C0B,>0C11 THRU 24
DATA >0C13,>0E07,>0E09,>0E13 THRU 30
DATA >1003,>1009,>100F,>1013,>1015 THRU 36
DATA >1203,>120D,>120F,>1213,>1407,>1409 THRU 42
DATA >140B,>140D,>1413,>1415,>1603 THRU 48

```

\*\*\* THERE ARE 40 CRATES TO THE SCREEN

```

MAZE3 DATA >0215,>0405,>0407,>0409 THRU 06
DATA >0411,>0415,>0417,>0609,>060B,>060D THRU 12
DATA >060F,>0611,>0811,>0815,>0A05 THRU 18
DATA >0A07,>0A09,>0A0F,>0A15,>0C05 THRU 24
DATA >0C0D,>0C13,>0E0B,>0E11 THRU 30
DATA >1003,>1007,>1009,>100F,>1015 THRU 36
DATA >1203,>1209,>120D,>1215,>1403,>1405 THRU 42
DATA >140D,>1411,>1413,>1415,>1609 THRU 48

```

\*\*\* THERE ARE 40 CRATES TO THE SCREEN

```

*****
* SOUND LISTS FOR VDPRAM *

```

\*\*\*\*\*

SLISTS BYTE 4, >9A, >BF, >DF, >FF, 1  
BYTE >03, >83, >15, >9A, >01  
BYTE >03, >82, >0E, >9A, >02 HOP SOUND  
BYTE >03, >80, >0A, >9A, >03 >1000 - >1016  
BYTE >03, >8B, >06, >9A, >01 29 BYTES  
BYTE >01, >9F, >00

\*  
BYTE 4, >9A, >BF, >DF, >FF, 1  
BYTE >03, >8B, >06, >94, >01  
BYTE >03, >8F, >05, >95, >01 CRATE SLIDE SOUND  
BYTE >03, >85, >05, >96, >01 >1017 - >103E  
BYTE >03, >8C, >04, >97, >01 49 BYTES  
BYTE >03, >83, >04, >98, >01  
BYTE >03, >8C, >03, >99, >01  
BYTE >03, >89, >03, >9A, >02  
BYTE >03, >85, >03, >98, >20  
BYTE 1, >9F, 0

\*  
DIELFT BYTE 3, >86, >0D, >97, >01 KANGO DIE LEFT >103F-1043  
DIERTGT BYTE 3, >8D, >11, >97, >01 KANGO DIE RIGHT >1044-1048

\*  
SNDAPR BYTE 4, >9A, >BF, >DF, >FF, 1 APPEAR SOUND  
BYTE 3, >87, >2A, >95, 2  
BYTE 1, >9F, 0 14

\*  
SNDMLT BYTE 4, >9A, >BF, >DF, >FF, 1  
BYTE >03, >8B, >23, >90, >03  
BYTE >03, >8B, >25, >91, >03 CHASER MELT SOUND  
BYTE >03, >8B, >27, >92, >03 >1017 - >103E  
BYTE >03, >8B, >2A, >93, >02 49 BYTES  
BYTE >03, >8B, >2C, >94, >02  
BYTE >03, >8B, >2D, >95, >02  
BYTE >03, >8B, >2E, >96, >01  
BYTE >03, >8B, >2F, >97, >01  
BYTE 1, >9F, 0

\*  
OFFSND BYTE 4, >9F, >BF, >DF, >FF, 0 SOUND OFF 5 BYTES

\*\*\*\*\*

\* INITIAL GAME PREPARATIONS BEFORE THE \*  
\* ACTUAL PROGRAM LOOP \*

\*\*\*\*\*

EVEN  
HOPPER LWPI MYWS USE OUR WORKSPACE  
LIMI 2 TURN ON INTERRUPTS

\*  
BL @CLEAR CLEAR THE SCREEN  
\*  
MOV B @H10, @DISINT DISABLE QUIT FUNCTION  
LI R1, >8320 START OF CPURAM  
CLRCPU MOV B @H00, \*R1+ ZERO OUT THAT BYTE  
CI R1, >8340 AT END?  
JL CLRCPU NO, SO KEEP LOOPING

\*  
MOV B @HE2, @SVVDP1 TO PRESERVE VDP R1  
BL @VWTR  
DATA >E281 DOUBLE SIZE SPRITES

\*  
LOADSD BL @WRTVCL  
DATA SNDBUF, 157, SLISTS

\*\*\*\*\*

\* THIS ROUTINE LOADS UP THE CHARACTERS \*  
\* THAT WILL BE USED FOR THE SCREEN. \*

\* RETURN LINKAGE IS IN R12

\*\*\*\*\*

LOADCH BL @WRTVCL                   GO PLACE THE TITLE CHARS  
DATA PDT, 160, TCHARS  
BL @WRTVCL  
DATA PDT->400, 166, ACHARS AUSTRALIA CHARS  
BL @WRTVCL  
DATA PDT->200, 8, CPYRIT

\*\*\*\*\*

\* THIS ROUTINE LOADS UP THE COLORS FOR CHAR  
\* THAT WILL BE USED ON THE SCREEN.       \*  
\* RETURN LINKAGE IS IN R12

\*\*\*\*\*

LOADCL LI R1, >9090  
BL @WRTVCL                   DO THE WALLS AND CRATES  
DATA CT, 2, MYWS+2            CORNERS YELLOW  
LI R1, >9000  
BL @WRTVCL                   MAKE THE KANGOS TRANS & LRED  
DATA CT+2, 1, MYWS+2            BLK  
LI R4, >A0A0  
LI R5, >A000  
BL @WRTVCL  
DATA CT->10, 3, MYWS+8        COLORS FOR AUSTRALIA

\*  
LI VDPADD, CT+4       START OF CHAR SET 4  
LI R1, >F000

LCLOOP BL @WRTCL           WRITE OUT THE COLOR  
DATA 1, MYWS+2       POINT TO CHAR SET 4  
INC VDPADD           NEXT CHARACTER SET  
CI VDPADD, >03BC     AT THE END OF LETTERS, #'S?  
JL LCLOOP            NO, SO KEEP GOING

\*\*\*\*\*

\* LOAD THE SPRITE DESCRIPTOR LIST       \*  
\* RETURN IS IN REGISTER 12.            \*

\*\*\*\*\*

LOADST BL @WRTVCL           WRITE OUT SPRITE PATTERNS  
DATA SDT, 224, FACE  
TITLES BL @CLEAR            CLEAR THE SCREEN  
BL @VWTR                    GO WRITE TO REGISTER  
DATA >4487                  CHANGE SCREEN COLOR  
BL @WRTVCL  
DATA SIT->285, 22, PRESSM  
BL @WRTVCL  
DATA SIT->2E4, 24, COPYM  
BL @WRTVCL  
DATA SIT->60, 64, TITSIT BIG HOPPER LETTERS

\*  
LI R6, 5                    LOOP COUNTER  
LI R4, TIT AUS            AUSTRALIA SIT CHARS  
LI R5, SIT->ED            CENTER FOR THE MAP

AUSLP LI WCOUNT, 6  
MOV R5, VDPADD            SET ADDRESS OF WHERE TO GO  
MOV R4, WLOC              SET ADDRESS OF WHAT TO WRITE  
BL @WRITE                 AND WRITE IT  
AI R5, >20                NEXT ROW  
AI R4, 6                  NEXT ROW OF SIT CHARS  
DEC R6                    DONE WITH 5 ROWS, YET?  
JNE AUSLP                 NO, SO KEEP WRITING

\*  
BL @KILTIM  
DATA >FFFF                1/2 SEC DELAY  
CLR @KEYBRD               99/4 KEYBOARD



```

BL @SCANKY          LET'S GET A KEY
MOVW @H00, @TIMER  CLEAR TIMER
CLR R4              SECONDS COUNTER
CHK1 CLR @SCNTIM
BL @SCANKY
MOVW @STATUS, R3   HAS THERE BEEN A KEY PRESSED?
JEG NOHIT          NOT ACCORDING TO STATUS BYTE
*
MOVW @KEY, R1      YES, ACCORDING TO STATUS
SRL R1, 8
JMP HITWAT        SO FIND OUT WHAT
NOHIT MOVW @TIMER, R7
SRL R7, 8
CI R7, >0060
JL CHK1           1 SECOND HASN'T PAST
INC R4            1 MORE SECOND GONE
CI R4, 5          5 SECONDS GONE?
*
JHE DEMO          YES, SO START DEMO MODE
*
MOVW @H00, @TIMER NO, BUT RESET TIMER
JMP CHK1
*
HITWAT CI R1, >002A  ASTERISK?
JEG HITASK
CI R1, >0024      '*'?
JEG HITDOL
JMP GETPLY       WRONG KEY
*
HITDOL BL @CLEAR
BL @WRTVCL
DATA SIT+>60, 64, TITSIT
BL @WRTVCL
DATA SIT+>142, 20, E001
AI VDPADD, >40
BL @WRTCL
DATA 20, E002
AI VDPADD, >40   OUR CLAIM TO FAME
BL @WRTCL
DATA 20, E003
LI R9, 4
HDOLL BL @KILTIM
DATA >FFFF
DEC R9
JNE HDOLL
JMP TITLES
*
DEMO MOVW @H00, @HOWMNY DEMO MODE ALWAYS ONE PLAYER
CLR @CNTR        CLEAR COUNTER
CLR @DEM         CLEAR FLAG FOR DEMO MODE
B @LPDT          START DEMO MODE
*
HITASK BL @CLEAR    CLEAR SCREEN
BL @WRTVCL
DATA SIT+>60, 64, TITSIT
BL @WRTVCL
DATA SIT+>103, 22, ROUNDM
MOVW @H00, @GAMLV1 START WITH ZERO
MOVW @H00, @GAMLV2 START WITH ZERO
GETRND BL @SCANKY
MOVW @KEY, R1
SRL R1, 8

```

```

CI R1,>00FF
JEG GETRND
CI R1,>0030
JL GETRND
CI R1,>0039
JH GETRND
LI R2,>0030          ASCII BIAS
S R2,R1
MOVB @R1LB,@GAMLV1 NOW HAVE A GAME LEVEL
MOVB @R1LB,@GAMLV2

```

```

*
GETPLY LI R1,>0100
MOV R1,@DEM          SET FLAG FOR NORMAL PLAY
BL @CLEAR            GAME STARTS, SOMEHOW!!
BL @WRTVCL
DATA SIT+>60,64,TITSIT
BL @WRTVCL
DATA SIT+>142,22,ONEMSG
AI VDPADD,>40
BL @WRTCL            GET # OF PLAYERS
DATA 22,TWOMSG

```

```

*
GPLYLP CLR @SCNTIM      NO SCREEN BLANKING
BL @SCANKY           GET PLAYER #
MOVB @KEY,R3
SRL R3,8
CI R3,>00FF
JEG GPLYLP           NO KEY PRESSED

```

```

*
CI R3,>0031           '1' KEY
JL GPLYLP
CI R3,>0032           '2' KEY
JH GPLYLP

```

```

*
LI R1,>0030          ASCII BIAS
S R1,R3              NOW WE HAVE PLAYER #
DEC R3               0,1 . . . NOT 1,2
MOVB @R3LB,@HOWMNY AND IT'S IN CPU

```

```

*
LPDT BL @CLEAR
BL @WRTVCL
DATA PDT,48,ULCORN  CREATE SCREEN CORNERS
BL @WRTVCL
DATA PDT+64,8,FILLER SCREEN FILLER
BL @WRTVCL
DATA PDT+96,32,CRATES HERES THE CRATE
BL @WRTVCL
DATA PDT+>80,32,LKANG

```

```

*
LI R1,>1AA1
BL @WRTVCL           COLORS FOR
DATA CT,2,MYWS+2     THOSE NEW
LI R1,>F000           CHARS
BL @WRTVCL
DATA CT+2,1,MYWS+2

```

```

*
BL @WRTVCL           ZERO OUT HIGH SCORE
DATA SIT+>119,6,ZSCORE

```

```

*****
* THIS ROUTINE FORMATS THE SCREEN USING *
* THE HCHAR AND VCHAR ROUTINES, MOSTLY. *

```

\* RETURN LINKAGE IS IN R12

\*\*\*\*\*

GDGAME BL @HCHAR  
DATA >0102,>0001 ULCORN  
BL @HCHAR  
DATA >1802,>0201 LLCORN  
BL @HCHAR  
DATA >0119,>0101 URCORN  
BL @HCHAR  
DATA >1819,>0301 LRCORN  
BL @VCHAR  
DATA >0202,>0516 LEFT WALL  
BL @VCHAR  
DATA >0219,>0516 RIGHT WALL  
BL @HCHAR  
DATA >0103,>0416 TOP WALL  
BL @HCHAR  
DATA >1803,>0416

PUTMSG MOVB @HOWMNY,@HOWMNY  
JNE PMFOR2  
BL @WRTVCL  
DATA SIT+>39,6,SCOREM  
JMP NOTWOP

PMFOR2 BL @WRTVCL  
DATA SIT+>39,6,SCORE1  
BL @WRTVCL 2, SO PUT HIS SCORE UP, TOO  
DATA SIT+>99,6,SCORE2

NOTWOP BL @WRTVCL  
DATA SIT+>F9,6,HIGHM  
BL @WRTVCL  
DATA SIT+>159,6,TIMEM

\*\*\*\*\*

\* THIS ROUTINE DELETES THE TWO SCORE AREAS\*  
\* OF THE SCREEN. RETURN IS IN R12. \*

\*\*\*\*\*

NOSCOR BL @WRTVCL  
DATA SIT+>59,6,ZSCORE  
MOVB @HOWMNY,@HOWMNY ARE TWO PLAYERS PLAYING?  
JEG NO2SCR NO, SO DON'T ZERO PLAYER 2

\*  
BL @WRTVCL  
DATA SIT+>B9,6,ZSCORE

\*  
NO2SCR MOVB @H05,@KANGS1 START WITH 5 KANGOS  
MOVB @H05,@KANGS2  
MOVB @H01,@WHICH1 START WITH PLAYER 1  
LI R10,>005E AND SCORE FOR PLAYER 1  
MOV R10,@SCADDR  
MOVB @GAMLV1,@GAMLV1  
JEG STARTG NOT FROM CHEAT MODE!!  
SB @H01,@GAMLV1 WILL BE INCREMENTED RIGHT AWAY!  
SB @H01,@GAMLV2  
JMP NEWLVL

STARTG MOVB @HFF,@GAMLV1 START WITH -1  
MOVB @HFF,@GAMLV2

\*\*\*\*\*

\* M A I N P L A Y I N G L O O P \*  
\* \* \*

\*\*\*\*\*

NEWLVL MOVB @HOWMNY,@HOWMNY ARE THERE 2 PLAYERS?  
JEG NLFOR1 NO, SO KEEP ON WITH ONE  
CB @WHICH1,@H01 IS IT PLAYER 1'S TURN?

```

        JEQ NLFOR1          YES, SO DO PLAYER 1 STUFF
*
NLFOR2 AB @H01,@GAMLV2    INCREMENT PLAYER2 GAME LEVEL
      CLR R1              PREPARE FOR MOVE
      MOVB @GAMLV2,R1     MOVE TO R1
      CI R1,>0A00         PAST 10TH LEVEL?
      JL NLGOON          NO, SO KEEP ON
      LI R1,>0400         YES, SO SET BACK TO FIFTH
      MOVB R1,@GAMLV2
      JMP NLGOON
*
NLFOR1 AB @H01,@GAMLV1    INCREMENT PLAYER 1 GAME LEVEL
      CLR R1
      MOVB @GAMLV1,R1     CURRENT GAME LEVEL
      CI R1,>0A00         ARE WE PAST LEVEL 10?
      JL NLGOON
      LI R1,>0400         BACK TO LEVEL 5
      MOVB R1,@GAMLV1
*
NLGOON SRL R1,7           AND SAVE DELAY COUNTER
      MOV @CHADLY(R1),@SAVDLY
      SRL R1,1           FOR COLORS
      MOV R1,R6
*****
* AT LEVEL 9 CHANGE CRATE COLOR TO TRANS *
*****
TRANS CI R1,>09          IS IT LEVEL 9??
      JNE CCRATE        NO, GIVE IT A COLOR
NCRATE BL @WRTVCL
      DATA CT+1,1,H11  MAKE IT BLACK ON BLACK
      JMP TRANZO
CCRATE LI R4,>A100       YELLOW ON BLACK
      BL @WRTVCL
      DATA CT+1,1,MYWS+8
TRANZO EQU *
*****
* DISPLAY GAME LEVEL ON SCREEN *
*****
      MOV R1,R2          COPY LEVEL #
      INC R2             ADJUST FOR ASSEMBLY
      CLR R1
      LI R0,10
      DIV R0,R1          ANSWER IN 1, REM IN 2
      AI R1,>30          ADD ASCII BIAS
      AI R2,>30          SAME HERE
      SWPB R1           10'S IN HB
      MOVB @R2LB,@R1LB  1'S IN LB
      BL @WRTVCL
      DATA SIT+>17B,2,MYWS+2  WRITE TO SCREEN
*
      LI R1,>0087
      MOVB @BCOLOR(R6),R1
      BL @VWTRG
*****
* DISPLAY # OF KANGOS ON THE SCREEN *
*****
SHOWK MOVB @HOWMNY,@HOWMNY ARE THERE 2 PLAYERS?
      JEQ SHOWK1        NO, SO SHOW 1'S KANGS
      CB @WHICH1,@H01   IS IT PLAYER 1'S TURN?
      JEQ SHOWK1        YES, SO SHOW HIS KANGS
*
SHOWK2 MOVB @KANGS2,R4    NO, SO SHOW 2'S

```

```

        JMP SHOWKC
*
SHOWK1 MOVB @KANGS1,R4      CURRENT # OF KANGOS
SHOWKC SRL R4,8
        DEC R4              SAVKAN
        CLR R5              DISKAN
        LI VDPADD,>1BB     SIT ADDRESS OF KANGOS
SKLOOP INC R5              COUNTER
        CI R5,5            DONE WITH THIS LOOP?
        JEQ SKOUT         YES
        C R5,R4          PAST CURRENT # OF KANGOS?
        JLE SHOWEM       NO, SO SHOW A KANGO
NOSHOW BL @SHOWCH
        DATA >2020,>2020  SHOW BLANKS
        JMP NXTROW       AND INCREMENT VDPADD
SHOWEM BL @SHOWCH
        DATA >1012,>1113  SHOW A KANGO
NXTROW AI VDPADD,>40      POINT TWO ROWS DOWN
        JMP SKLOOP       AND KEEP GOING
*
* ROUTINE TO SHOW CHARACTERS ON SCREEN
*
SHOWCH MOV *R11+,R8        TOP TWO CHARS
        MOV *R11+,R9      BOT TWO CHARS
        MOV R11,R12      SAVE RETURN
*
        BL @WRTCL
        DATA 2,MYWS+16  TOP TWO
        AI VDPADD,>20    NEXT ROW DOWN
        BL @WRTCL
        DATA 2,MYWS+18  BOT TWO
        B *R12          RETURN TO CALLER
*
SKOUT  EGU $
*****
* FILLS PLAYING AREA WITH CHAR 8 *
*****
FILBLK LI R8,>0200        ROW 2, COL 3
FLBKLP LI R5,>0300
        MOV R8,R4
        BL @HCC          ROW, COL ALREADY LOADED
        DATA >0816     26 ACROSS
        AB @H01,R8      NEXT ROW
        CI R8,>1700     PAST ROW 23?
        JLE FLBKLP     NO, SO KEEP GOING
*****
* THIS ROUTINE WILL PUT CRATES ON SCREEN. *
*****
PUTCRT MOVB @HOWMNY,@HOWMNY  HOW MANY PLAYERS?
        JEQ ONLY1        ONE
        CB @WHICH1,@H01  TWO, BUT WHICH ONE?
        JNE ONLY2        PLAYER 2
ONLY1  MOVB @GAMLV1,R8      USE PLAYER 1'S LEVEL
        JMP ONLYME
ONLY2  MOVB @GAMLV2,R8      USE PLAYER 2'S LEVEL
ONLYME SRL R8,8            GAME LEVEL INTO LSB
        CI R8,2          AT FOURTH LEVEL?
        JLE PMAZE1      NO, SO USE MAZE 1
        CI R8,5          AT SEVENTH LEVEL?
        JLE PMAZE2      NO, SO USE MAZE 1
        JMP PMAZE3      USE LAST MAZE
*

```

```

PMAZE1 LI R8,MAZE1          ADDRESS OF FIRST MAZE
      JMP PMCONT
PMAZE2 LI R8,MAZE2          ADDRESS OF SECOND MAZE
      JMP PMCONT
PMAZE3 LI R8,MAZE3
PMCONT LI R1,40             FORTY CRATES TO A SCREEN
*
PCLOOP MOVB *R8+,R4         HCHAR DICTATES THAT ROW,COL
      MOVB *R8,R5           BE IN THE MSB OF THE REGS
      DEC R8                POINT BACK TO ROW BYTE
      BL @HCC               LETS GO PUT A CRATE OUT
      DATA >0C01          CHAR AND COUNT
*
      MOVB *R8+,R4         HCHAR DICTATES THAT ROW,COL
      MOVB *R8,R5           BE IN THE MSB OF THE REGS
      DEC R8                POINT BACK TO ROW BYTE
      AI R4,>0100          LL CRATE
      BL @HCC               LETS GO PUT A CRATE OUT
      DATA >0D01          CHAR AND COUNT
*
      MOVB *R8+,R4         HCHAR DICTATES THAT ROW,COL
      MOVB *R8,R5           BE IN THE MSB OF THE REGS
      DEC R8                POINT BACK TO ROW BYTE
      AI R5,>0100          UR CRATE
      BL @HCC               LETS GO PUT A CRATE OUT
      DATA >0E01          CHAR AND COUNT
*
      MOVB *R8+,R4         HCHAR DICTATES THAT ROW,COL
      MOVB *R8+,R5         BE IN THE MSB OF THE REGS
* NOTE==>                  ** R8 NOW POINTS TO NEXT DATA PAIR
      AI R4,>0100          LR CRATE
      AI R5,>0100
      BL @HCC               LETS GO PUT A CRATE OUT
      DATA >0F01          CHAR AND COUNT
*
      DEC R1                DONE WITH CRATES?
      JNE PCLOOP           NO, SO DO IT AGAIN!!
*
*****
* PUT 3 CHASERS OUT ON THE SCREEN *
* ALSO ZEROES THEIR TIMERS AND INITS *
* KANGO'S SIT ADDRESS. *
*****
PUTSPR BL @WRTVCL          PUT ALL SPRITES IN SAL
      DATA SAL+4,30,CRTINI
      CLR @CH1TIM          ZEROES THERE
      MOVB @H00,@CHOTIM AND ONE THERE!
*
      LI R1,>8C00
      MOVB R1,@KCHAR        LEFT FACING CHAR
      LI R1,>01AC          KANGO SIT ADDRESS
      MOV R1,@KPOS
      BL @WRTVCL
      DATA SAL,1,HCO      NULLIFY KANGOROO MOMENTARILY
*
      MOVB @H00,@CHAKIL NO CHASERS KILLED IN NEW LEVEL
      MOVB @H00,@KEYDLY ZERO KANGO MOVE DELAY COUNT
*
      LI R0,>FFFF          1/2 SECOND
WAITJY DEC R0              GIVE THE PLAYER A CHANCE
      JNE WAITJY
*

```

```

BL @SOUND
DATA APRSND          SOUND HIS ROYAL APPEARANCE
BL @WRTVCL
DATA SAL,4,KANINI PUT KANGAROO OUT NOW

```

```

*
WAITJZ LI R0,>4000
DEC R0
JNE WAITJZ

```

```

*
KANGOL CLR @SCNTIM      NO SCREEN TIME-OUT ALLOWED
BL @CKANG      CHECK JOYSTICKS/MOV KANGO
BLWP @MOVCHA   TIME TO MOVE CHASERS?
CB @CHAKIL,@H03 ALL THE CHASERS GONE?
JNE KANGOL     NO, SO KEEP @DING

```

```

*
GHLOOP LI R4,4          YES, SO WAIT FOR NEXT LEVEL
LI R1,>1F1A          FLASHING COLORS
BL @WRTVCL
DATA CT,1,MYWS+2   FLASH WHITE
BL @KILTIM
DATA >8000
BL @WRTCL
DATA 1,R1LB        RESTORE ORIGINAL COLOR
BL @KILTIM
DATA >8000
LI R10,>250
BL @SCORE          ADD 1000 POINTS
DEC R4             DONE WASTING TIME?
JNE GHLOOP        NOP
B @NEWLVL         YES, SO NEW LEVEL

```

```

*****
*
* ROUTINE TO TEST KEYBOARD AND JOYSTICKS *
* RETURN LINK IN R14 *
*****

```

```

CKANG MOV R11,R14
MOV B @KEYDLY,R1 CAN WE MOVE, YET?
SRL R1,8
INC R1
MOV B @R1LB,@KEYDLY SAVE DELAY COUNTER
MOV B @HOWMNY,@HOWMNY HOW MANY PLAYERS ARE THERE?
JEG MONLY1 ONLY ONE PLAYER
CB @WHICH1,@H01 IS IT PLAYER ONE'S TURN?
JEG MONLY1 YES, SO USE ONLY GAMLVL1 TABLE
MOV B @GAMLV2,R2 PLACE LEVEL IN R0.***I HOPE ITS FREE!!***
SRL R2,8 PLACE IN LOW BYTE
MOV B @KDELAY(R2),R0 PLACE LEVEL DELAY IN R0
JMP TESTIT TEST LEVEL DELAY
MONLY1 MOV B @GAMLV1,R2 PLACE PLAYER ONE LEVEL IN R0
SRL R2,8 PLACE IN LOW BYTE
MOV B @KDELAY(R2),R0 PLACE P1 LEVEL DELAY FOR TEST
TESTIT SRL R0,8 PUT DELAY INTO LSB JUST LIKE R1
C R1,R0 TEST DELAY
JHE NOW CHECK WHOLE KEYBOARD
CLR R1 SET FLAG FOR CHECK FIRE ONLY
JMP LATER
NOW CLR @KEYDLY TWO WITH ONE SHOT!!!!
LATER CLR @KEYBRD CHECK ONLY FIRE KEY
BRDINC MOV B @DEM,@DEM ARE WE IN DEMO MODE
JNE NPLAY NORMAL PLAY
AB @H01,@DEMDLY
BL @SCANKY

```

```

MOV B @STATUS, @STATUS   KEY PRESSES
JNE DEMBCK                NO, GO BACK TO TITLES
MOV B @DEMDLY, R1
SRL R1, 8
CI R1, >10                TIME TO MOVE?
JL DEMOUT
MOV B @H00, @DEMDLY
DEMCN C @DEM, @CNTR      ENOUGH OF JUMPING
      JH NORND
      AB @H01, @FLAG
      CB @FLAG, @H03     THROUGH BRDINC LOOP 2 TIMES
      JNE GRND
      MOV B @H00, @FLAG  CLEAR FLAG
DEMCN CLR @CNTR          CLEAR COUNTER
      B @INPTRT         GO BACK
GRND  BL @RANDNO
      SLA R10, 8
      SRL R10, 13        GET A 3 BIT NO.
      CI R10, 4          IS R10<4
      JH GRND           YES GO BACK TO RANDNO
      SLA R10, 1        MPY BY 2
ARND  MOV @RNDJMP(R10), @WWAY WHAT DIRECTION SHALL WE GO
      BL @RANDNO
      SLA R10, 8
      SRL R10, 13        GET ANOTHER RNDNO
      CI R10, 6          R10<6
      JH ARND           NO DO ARND AGAIN
      MOV R10, @DEM
NORND INC @CNTR
      MOV @WWAY, R1
      B *R1              JUMP AGAIN
DEMBCK LI R1, >D000
      BL @WRTVCL
      DATA SAL, 1, MYWS+2
      B @HOPPER
*
NPLAY AB @H01, @KEYBRD
      CB @KEYBRD, @H03
      JNE NPLAYC
      B @INPTRT         RETURN TO CALLER
NPLAYC BLWP @MOVCHA      ADDED FOR TESTING PURPOSES
      BL @ESCANKY
      CLR R3            GOING TO PUT KEY IN HERE
      MOV B @KEY, R3
      MOV R1, R1        IS R1=0
      JEQ FIPA         YES, CHECK ONLY FIRE KEYS
      CB @KEY, @HFF     NO, CHECK WHOLE THING
      JEQ CHKKEY
CHKKEY CI R3, >0500
      JGT FIPA
      MOV B @KEYDLY, @KEYDLY ARE WE ALLOWED TO MOVE?
      JEQ KEYOK        YES, SINCE @KEYDLY IS CLEARED
      LI R3, >0100     NO, SO SET BRANCH BACK TO @BRDINC
KEYOK SWPB R3
      SLA R3, 1
      MOV @WHERE(R3), R10
      B *R10
FIPA  CB R3, @H12
      JEQ FIRE
      CB R3, @H0D
      JEQ FIRE
      CB R3, @H0B

```



```

JEG PAUSE
JMP BRDINC
CHKJOY C @JOYY, @H0000
JEG BRDINC
CB @JOYY, @H00
JGT MUP DISPLACEMENT TOO BIG
JLT DOWN
CB @JOYX, @H00
JGT RIGHT
LEFT LI R6, >8C00
LI R8, >FFFF
LI R9, >FFFC
MOV R8, R15
JMP JMPPCH DISPLACEMENT TOO BIG
RIGHT LI R6, >9000
LI R8, >0002
LI R9, >0004
LI R15, >0001
JMP JUMP
FIRE B @KICK
MUP JMP UP
JMPPCH JMP JUMP
*****
* THIS MODULE IS THE PAUSE FEATURE INCLUDED IN ALL T. I. GAMES. *
* IT WILL STAY HERE UNTIL ANY KEY IS HIT AGAIN *
*****
PAUSE MOVB @MOTION, R2 SAVE MOTION BYTE
MOVB @H00, @MOTION AND SET TO NO MOTION
MOVB @KEYBRD, R3 SAVE WHICH KEYBRD IT WAS IN
BL @WRTVCL
DATA SIT+>2EA, 6, PAUSEM PAUSE MSG
MOVB @H01, @PAUSEF AND MSG IS ON THE SCREEN
*
PAUSL1 BL @SCANKY GO CHECK FOR SOME INPUT
CB @KEY, @HFF KEY STILL PRESSED (DEBOUNCE?)
JNE PAUSL1 YES, SO WAIT UNTIL HE LETS UP
CLR @PAUSET CLEAR THE TIMER
*
PAUSL2 CLR @SCNTIM CLEAR SCREEN TIME-OUT
INC @PAUSET INCREMENT TIMER
MOV @PAUSET, R0
CI R0, >0100 TIMER UP?
JLE P2KEY NO, SO SCAN FOR KEY
*
CLR @PAUSET CLEAR THE TIMER AGAIN
MOVB @PAUSEF, @PAUSEF IS MSG ON SCREEN?
JNE CLRMSG YES, SO ERASE IT
*
MOVB @H01, @PAUSEF NO, BUT IT WILL BE
BL @WRTVCL THERE IT IS
DATA SIT+>2EA, 6, PAUSEM AND NOW CHECK FOR THE KEY
JMP P2KEY
*
CLRMSG MOVB @H00, @PAUSEF NO MORE MSG
BL @WRTVCL
DATA SIT+>2EA, 6, PAUSFL PUT A FILLER BACK IN
*
P2KEY CLR @KEYBRD NOW GO CHECK FOR A KEY
BL @SCANKY KEY PRESSED?
MOVB @STATUS, R0 NO, SO CHECK JOYSTICKS
JNE PAUSL3
*

```

```

P2JOY  MOVB @H01,@KEYBRD
        BL @SCANKY
        CB @KEY,@HFF
        JNE PAUSL3
        MOVB @H02,@KEYBRD
        BL @SCANKY
        CB @KEY,@HFF
        JNE PAUSL3
        JMP PAUSL2

```

```

*
PAUSL3  BL @SCANKY
        CB @KEY,@HFF
        JNE PAUSL3
        MOVB R2,@MOTION
        MOVB R3,@KEYBRD
        BL @WRTVCL
        DATA SIT->2EA,6,PAUSFL
        B @LATER

```

NOW WAIT UNTIL HE RELEASES THAT ONE  
 RESTORE MOTION  
 RESTORE KEYBRD  
 MAKE SURE PAUSE MSG IS ERASED!  
 LET'S GO ON WITH THE GAME

```

*****
* RETURN POINT FOR MOVING SUBROUTINES*
*****
INPTRT BLWF @MOVCHA
        B *R14

```

```

*
DOWN   LI R6,>9800
        LI R8,>40
        LI R9,>0004
        LI R15,>20
        JMP JUMP
UP     LI R6,>9400
        LI R8,>FFE0
        LI R9,>FFFC
        MOV R8,R15

```

```

*****
*
* THIS ROUTINE MOVES KANGO IN THE DIRECTION *
* PASSED BY THE JOYSTICK/KEYBOARD READ *
* ROUTINES. THE CHARACTER CODE OF THE NEW *
* DIRECTION SHOULD BE PLACED IN R6 IN THE *
* FOLLOWING FORMAT: >STANDINGCHARJUMPINGCHAR*
* I.E. >8084. THE POINTER FOR U/D/L/R SHOULD*
* BE LOADED INTO R9 IN THE FOLLOWING FORMAT:*
* >0004 IF THE MOTION IS TO BE DOWN OR RIGHT*
* >FFFC IF THE MOTION IS TO BE UP OR LEFT *
*****

```

```

*
JUMP   CLR R5
        MOV @KPOS,R7
        MOV R7,R13
        CB R6,@KCHAR
        IS MOTION IN SAME DIR
        AS PREVIOUS?
*
        JEQ RPT
        YES,EXECUTE JUMP ROUTINE
        BL @WRTVCL
        NO,WRITE NEW DIR CHAR
        DATA SAL+2,1,MYWS+12
        MOV R6,@KCHAR
        UPDATE KANGO POSITION
        B @JRT
        RETURN TO CALLING RTN
RPT    CI R6,>9200
        JH UDJUMP
        GOTO RTN FOR U/D MOVE
RLJUMP A R8,R7
        INC/DEC POSITION CNTR
        MOV R7,VPADD
        BL @RDCL
        GET CHAR IMM TO THE L/R
        DATA 1,MYWS+8
        OF THE CHAR

```

```

AI    VDPADD, >20    INC CNTR TO NEXT ROW
BL    @RDCL          READ CHAR 1 ROW DOWN
DATA  1, R4LB
LI    R8, 1          SET CNTR FOR L/R JUMP
CI    R4, >0808     ARE BOTH CHARS FILLER?
JEG   HOP            YES, EXECUTE HOP
JNE   JRT            NO, RTN TO CALLER
UDJUMP A   R8, R7     INC/DEC POS CNTR
      MOV   R7, VDPADD
      BL    @RDCL     GET CHARS
      DATA 2, MYWS+8
      CLR   R8        SET CNTR FOR U/D JUMP
      CI    R4, >0808 ARE BOTH CHAR FILLER?
      JNE   JRT        NO, RTN TO CALLER
HOP   INC   R5        INC JUMP CNTR
      CLR   R4
      BLWP @MOVCHA    KILL TIME BEFORE JUMP
      AI    R8, >300
      MOV   R8, VDPADD
      BL    @RDCL     READ X/Y POS OF
      DATA 1, R4LB   KANGO
      A     R9, R4     ADJUST X/Y VALUE FOR MOVE
      BL    @WRTCL    WRITE OUT NEW POS
      DATA 1, R4LB
      CI    R5, 02     HAS KANGO GONE 8 PIXELS
      JEG   URT        YES, RTN TO CALLER
      BLWP @MOVCHA    KILL TIME BEFORE NEXT JUMP
      AI    R8, >FD00  SUBTRACT >300 TO RESTORE ORIGINAL POINTER
      JMP   HOP        NEXT JUMP
URT   A     R15, R13
      MOV   R13, @KPOS  UPDATE KANGO POS
      MOVB  R6, @KCHAR  UPDATE KANGO CHAR
      BL    @SOUND     MAKE BLEEP SOUND
      DATA HOPSND
JRT   B     @BRDINC
*****
* WASTE SOME TIME BETWEEN MOVES!*
*****
KILTIM MOV  *R11+, R10
KIL    DEC  R10
      JNE  KIL
      B   *R11
*****
* KICK ROUTINE. BRANCHED OFF OF CKANG *
*****
KICK   MOVB @KCHAR, R4  LOAD CHARACTER POINTER INTO R4
      MOV  @KPOS, R6    LOAD POSITION POINTER INTO R6
      SRL  R4, 8        SHIFT CHAR POINTER TO R4LB
      CI   R4, >0092    IS KICK UP OR DOWN?
      JH   KICKUD       YES, CHECK UP/DOWN BNDRY
KICKLR LI   R5, 2       LOAD SIT OFFSET FOR BNDRY CHECK
      CI   R4, >008E    IS KICK TO THE LEFT?
      JH   CHECK1       YES, CHECK LEFT BNDRY
KICKL  LI   R5, >FFFF   NO, CHECK RIGHT BNDRY
CHECK1 MOV  R6, VDPADD  ADD SIT OFFSET TO R6, CHARACTER TO CHECK
      A    R5, VDPADD  LOAD CHARACTER POS INTO VDPADD
      BL   @RDCL       READ CHAR FR SIT
      DATA 1, MYWS+2  INTO R1
      AI   VDPADD, >20 CHECK NEXT ROW
      BL   @RDCL       READ NEXT CHAR FR SIT

```

```

DATA 1, MYWS+3      INTO R1LB
CI R1, >OCOD        IS IT A LEFT CRATE?
JEG PUNCH           YES, THEN PUNCH IT
CI R1, >OEOF        IS IT A RIGHT CHAR?
JEG PUNCH           YES, THEN PUNCH IT
B @BRDINC           NO, GO BACK TO CALLER
KICKUD LI R5, >FFEO  LOAD POINTER FOR UP
CI R4, >O096        IS IT AN UP CHAR?
JL CHECK2           YES, CHECK UP BNDRY
KICKD  LI R5, >40    NO, CHECK DOWN BNDRY
CHECK2 MOV R6, VDPADD ADD SIT OFFSET TO R6, CHARACTER TO CHECK
A R5, VDPADD        LOAD CHARACTER POS INTO VDPADD
BL @RDCL            READ TWO CHAR FR SIT
DATA 2, MYWS+2      INTO R1
CI R1, >OCOE        IS IT AN UP CHAR?
JEG PUNCH           YES, THEN PUNCH IT
CI R1, >ODOF        IS IT A DOWN CHAR?
JEG PUNCH           YES, THEN PUNCH IT
B @BRDINC

```

\*\*\*\*\*

\* THIS ROUTINE WILL PLACE A CRATE \*  
\* SPRITE ON THE SCREEN AND DELETE THE \*  
\* CHARACTER CRATE. \*

\*\*\*\*\*

```

PUNCH BL @RDVCL
DATA SAL, 2, MYWS+2 GET SAL OF KANGO INTO R1
LI R8, >F000        UP TWO CHARACTERS
LI R9, >FFC0        UP TWO SCREEN CHARS
LI R7, >FF00        CRATE INCREMENT
CI R4, >94          UP KICK?
JEG PUNCHO         YES, SO GO PUNCH IT OUT
SRA R7, 8          LEFT TWO CHARACTERS
LI R9, -2          LEFT TWO SCREEN CHARS
SRA R8, 8          CRATE DIR INCREMENT
CI R4, >9C          LEFT KICK?
JEG PUNCHO         YES, SO GO PUNCH IT OUT

```

\*  
\*

```

LI R8, >0010        RIGHT TWO CHARACTERS
ABS R9             RIGHT TWO SCREEN CHARS
ABS R7             CRATE INCREMENT
CI R4, >90          LEFT KICK?
JEG PUNCHO         YES, SO GO PUNCH IT OUT

```

\*

```

SLA R8, 8          DOWN TWO CHARACTERS
SLA R9, 5          DOWN TWO SCREEN CHARS
SLA R7, 8          CRATE INCREMENT

```

\*

```

PUNCHO LI R10, 10   10 POINTS FOR A KICK
BL @SCORE         SET FLAG FOR EXPLODE
LI R12, 1         ADD POS INCREMENT
A R8, R1          CRATE SPRITE POSITION
LI VDPADD, >0304
BL @WRTCL
DATA 2, MYWS+2
A R9, R6          SIT OFFSET
LI R1, >0808       FILL CHARS
MOV R6, VDPADD    LOAD VDPADD
BL @WRTCL
DATA 2, MYWS+2
AI VDPADD, >20     NEXT ROW DOWN
BL @WRTCL

```

```

DATA 2, MYWS+2
*
LI   VDPADD, >308 POINT TO FIRST CHASER
CLR  R4
GETCHA BL @RDCL READ CHASER POSITION(S)
DATA 2, MYWS+2 INTO R1
MOV  R1, @CHAPOS(R4) PLACE VALUE IN CHASER POSITION STACK
INCT R4
AI   VDPADD, 8 POINT TO NEXT CHASER
CI   R4, 6 3 CHASERS DONE
JNE  GETCHA NO, GET ANOTHER ONE
BL  @SOUND
DATA SLDSND EXECUTE SLIDE SOUND
BLOCK MOVB @KCHAR, R4 READ KICK DIR INTO R4
SRL  R4, 8 SHIFT TO LOW BYTE
A    R5, R6 ADD SIT OFFSET FOR BNDRY CHK
MOV  R6, VDPADD INTO VDPADD
CI   R4, >0092 IS KICK UD OR LR
JH   BLOCUD KICK IS UP
BLOCLR BL @RDCL GET CHAR FOR LR BNDRY CHK
DATA 1, MYWS+26 INTO R13
AI   VDPADD, >20 CHECK NEXT ROW
BL  @RDCL GET NEXT CHAR INTO R1LB
DATA 1, R13LB
CI   R13, >0808 ARE CHARS FILLER
JNE  BLOCKD NO, STOP IT
JMP  CRATMV YES, MOVE CRATE
BLOCUD BL @RDCL GET CHARS FOR UD BNDRY CHECK
DATA 2, MYWS+26 INTO R13
CI   R13, >0808 ARE CHARS FILLER?
JEG  CRATMV YES, MOVE CRATE
BLOCKD MOV R12, R12 WAS CRATE BLOCKED WHEN FIRST KICK?
JNE  EXPLOD YES, SO EXPLODE
B    @CRTSTP NO, SO STOP IT
EXPLOD BL @SOUND
DATA SNDOFF MAKE THE CRASHING SOUND
LI   VDPADD, CT+1
BL  @RDCL FLASH THE CRATES
DATA 1, MYWS+24
CB  R12, @H11 ARE CRATES TRANSPARENT?
JNE  EXCONT NO, SO GO AS NORMAL
BL  @WRTCL YES, SO FLASH
DATA 1, HA1
*
EXCONT LI VDPADD, >0307 YES, EXPLODE IT
LI   R6, >4000
LI   R1, 4 PLACE COLOR IN R1
EXPLOD BL @WRTCL
DATA 1, R1LB WRITE COLOR
MOV  R6, R5
KT   DEC R5
JNE  KT
INC  R1 POINT TO NEXT COLOR
SRL  R6, 1 DIVIDE DELAY BY TWO
CI   R1, >8 HAS IT GONE THROUGH 7 COLORS?
JNE  EXPLOD NO, DO IT AGAIN
LI   VDPADD, >0304 SAL POS OF CRATE Y COR
BL  @WRTCL WRITE NEW CRATE Y COR
DATA 1, HCO NEW Y COR IS OFF SCREEN BOTTOM
LI   R10, >50
BL  @SCORE
*

```

```

CB R12,@HA1      ARE CRATES ON?
JEQ KRGO         YES, SO DON'T ERASE
LI VDPADD,CT+1
BL @WRTCL
DATA 1,H11

*
KRGO  B    @KIKRTN      RETURN TO CALLER
*
CRATMV S    R5,R6      RESTORE VALUE
*

CLR  R4          CLR COUNTER
CLR R12         CLR EXP/STOP FLAG
LI   VDPADD,>0304    POINT TO CRATE Y, X POS
BL   @RDCL       READ CRATE Y, X
DATA 2,MYWS+2    INTO R1
CRATM A    R7,R1      ADD INC TO R1
BL   @WRTCL      MOV CRATE 1 PIXEL
DATA 2,MYWS+2
INC  R4          INC COUNTER
CI   R4,16       MOVED 16 PIXELS
JEQ  CCOINC      YES, CHECK COINC
BL   @KILTIM     NO, KILL TIME
DATA >90        BEFORE NEXT MOVE
JMP  CRATM       MOVE IT AGAIN
CCOINC A    R9,R6
MOV  R1,R10      COPY R1 INTO R10
SRL  R1,8        MOVE TO R1LB
ANDI R10,>00FF   CLEAR R10HB
CLR  R4          CLR CNTR
CK   MOVB @CHAPOS(R4),R8  MOVE CHAS. Y INTO R8
SRL  R8,8        MOV TO R8 LB
MOVB @CHAPOS+1(R4),R15  MOV CHAS X INTO R9
SRL  R15,8       MOV TO R9LB
S    R1,R8       SUB CRAT Y FROM CHAS Y
ABS  R8          TAKE ABS VALUE
S    R10,R15     SUB CRAT X FR. CHAS X
ABS  R15        TAKE ABS VALUE
CI   R8,8        ABS Y DIFF <8
JH   CHNEXT     NO, CHECK NEXT CHASER
CI   R15,8      ABS X DIFF <8
JH   CHNEXT     NO, CHECK NEXT CHASER
CRRYCH LI VDPADD,>308  POINT TO CHASER
AB   @H02,@BONUS  INCREMNET BONUS
MOV  R4,R15     COPY CNTR TO R15
SLA  R15,2      MUPLY BY 4
A    R15,VDPADD  HATS SAL POS
BL   @WRTCL     MOV HAT TO CO
DATA 1,HCO
C    *VDPADD+,*VDPADD+  ADD 4
AB   @H01,@CHAKIL  INC # OF CHASERS KILLED
BL   @WRTCL     MOV FACE TO CO
DATA 1,HCO
CHNEXT MOVB @HCO,@CHAPOS(R4)  UPDATES CPU LOCATION AS WELL
INCT R4
CI   R4,6       THREE CHASERS CHECKED?
JNE  CK        NO, CHECK NEXT ONE
B    @BLOCK     CHECK BNDRY AGAIN
CRTSTP BL @SOUND
DATA SNDOFF     TURN OFF SOUND GENERATOR
MOV  R1,R4
SLA  R4,8
DEC  R4

```

```

SRL R4,11
SLA R4,5
MOV R10,R6
SLA R6,8
SRL R6,11
A R6,R4 VOILA' THE CRATE SIT
AI R4,>20
LI R5,>0C0E TOP TWO CHAR OF CRATE
MOV R4,VDPADD POINT TO CRATE SIT
BL @WRTCL
DATA 2,MYWS+10
AI VDPADD,>20
AI R5,>0101
BL @WRTCL REPLACE SPRITE W/ GRAPHIC
DATA 2,MYWS+10
LI VDPADD,>0304 POINT TO SPRITE CRATE Y POS
BL @WRTCL
DATA 1,HCO MOVE CRATE OFF BOTTOM OF SCREEN
MOVB @BONUS,R1
SRL R1,8
MOV @POINT(R1),R10
BL @SCORE
KIKRTN CLR @BONUS
B @BRDINC
*****
* MOVE CHASERS ROUTINE. PASSES THROUGH 3 *
* TIMES. RETURN IS IN R15. *
*****
MOVCHA DATA MYWS2,MCCODE SAVE RETURN
MCCODE DEC @SAVDLY TIME TO LET CHASERS MOVE?
JEG GOMVCH YES, SO GO MOVE CHASERS
BL @COINCK NO, BUT STILL CHECK COINCIDENCE
B @MCRTRN AND SKIP REST OF CODE IF NO COINC
*
GOMVCH CLR R7
MOVB @HOWMNY,@HOWMNY ARE THERE TWO PLAYERS?
JEG MVFOR1 NO, SO MOVE FOR PLAYER 1
CB @WHICH1,@H01 ARE WE ON PLAYER 1?
JEG MVFOR1 YES, SO USE PLAYER 1 LEVEL
*
MVFOR2 MOVB @GAMLV2,R7 USE PLAYER 2 LEVEL
JMP GOMVCT AND CONTINUE
*
MVFOR1 MOVB @GAMLV1,R7 USE PLAYER 1 GAME LEVEL
GOMVCT SRL R7,7 EVEN MULT OF TWO!!!
MOV @CHADLY(R7),@SAVDLY
LI R7,SAL+8 POINT TO FIRST SPRITE
CLR R8 TABLE POINTER PER CHASER
***** NOTICE !!! R7 MUST NOT BE DESTROYED!!!!
MLOOP MOV R7,VDPADD SET UP VDPADD
BL @RDCL
DATA 1,MYWS2+2 GET HAT Y, X
SRL R1,8 INTO LSB
CI R1,>00C0 IS THIS SPRITE ON SCREEN?
JHE NXTCHA NO, SO GO ON TO NEXT CHASER
*
MOVB @CHATIM(R8),R12 TIME UP?
JEG SETTIM YES, SO NEW DIRECTION TIME!
*
MOVB @CHSVDR(R8),R12
SRL R12,8 NO, SO KEEP CURRENT DIR
SB @H01,@CHATIM(R8)

```

```

        JMP HAVDIR          AND CHECK TO SEE IF WE CAN
*
SETTIM  MOVB @HOA,@CHATIM(R8)  RESET CHANGE DIR TIMER

MCRAND  BL @RANDNO          GET A RANDOM #
        ANDI R10,>0300
        SRL R10,7           0,2,4 OR 6
        MOV R10,R12        SAVE IT
HAVDIR  MOVB @MYWS2+25,@RANINC  SAVE IT OUT
        SB @HO2,@RANINC    GOING TO STOP RECIPROCAL BOUNCING
        JGT GOTRI         STILL +
        JEQ GOTRI         AND STILL ZERO
        MOVB @HO6,@RANINC  WENT NEGATIVE, SO MAKE 6
GOTRI   SB @HO2,@RANINC    WILL BE INCREMENTED RIGHT AWAY
        MOVB @H00,@RANCNT  COUNTER FOR ENTRAPMENT
MCSET   MOV @WHERTO(R12),R10
        MOV @DIRINC(R12),R9  DIRECTION INCREMENT
        B *R10            GOTO DIR ROUTINE

*
**** UPON RETURN, R1 CONTAINS MOVE O.K. FLAG
MCCONT  MOV R1,R1          CAN WE MOVE IN THAT DIR?
        JNE MCGO         YES, SO MOVE IT BUSTER!!

*
        AB @HO1,@RANCNT    INCREMENT COUNTER
        AB @HO2,@RANINC    AND NEXT DIRECTION
        CB @RANINC,@HO6    PAST LIMITED DIRECTIONS?
        JLE NTOOHI        NO, SO IT'S NOT TOO HIGH
        MOVB @H00,@RANINC  TOO HIGH, SO SET TO ZERO
NTOOHI  CB @RANCNT,@HO5    HAVE WE EXHAUSTED OUR MOVES?
        JEQ CHTRAP        YES, SO CHASER IS TRAPPED!!
        CLR R12           FOR NEXT ROUTINES
        MOVB @RANINC,@MYWS2+25  WHER TO GO, WHAT TO MOVE
        JMP MCSET         TRY THE NEXT DIRECTION
MCGO    MOV R7,VDPADD      SET UP VDPADD
        BL @RDCL
        DATA 2,MYWS2+2    Y,X INTO R1
        A @CHADIR(R12),R1  INC/DEC POSITION
        BL @WRTCL
        DATA 2,MYWS2+2    AND REWRITE
        MOV R7,VDPADD
        C *VDPADD+,*VDPADD+  DO THE FACE, TOO
        BL @WRTCL
        DATA 2,MYWS2+2
        MOVB @MYWS2+25,@CHSVDR(R8)  RESAVE DIR POINTER

*
        MOV R1,R5          CONTAINS CHASER Y,X
        BL @COINCK        GO CHECK FOR COINCIDENCE

*
NXTCHA  INC R8            NEXT OFFSET INTO TABLES
        AI R7,8           POINT TO NEXT CHASER
        CI R7,>0320       PAST ALL CHASERS?
        JL MCLoop        NO, SO KEEP GOING

*
MCRTRN  RTWP

*
CHTRAP  BL @SOUND         NEED A MELTING SOUND NOW
        DATA MLTSND
        MOV R7,VDPADD      WHAT HAT
        LI R4,>6000        DELAY FOR FLASHING
        LI R1,7           HAT FLASH COLOR
        LI R6,6           FACE FLASH COLOR
        AI VDPADD,3       POINT TO HAT COLOR

```



```

FLSH  BL @WRTCL          CHANGE COLOR
      DATA 1,MYWS2+3    FROM R1LB OF WS2
      AI VDPADD, 4       POINT TO FACE
      BL @WRTCL          AND CHANGE COLOR
      DATA 1,MYWS2+13   FROM R6LB OF WS2
      MOV R4, R5         FOR KILTIM
KI    DEC R5
      JNE KI
      SRL R4, 1          DELAY/2
      INC R1             NEXT COLOR
      DEC R6            NEXT FACE COLOR
      AI VDPADD, >FFFC   POINT BACK TO HAT
      CI R1, >A         ALL COLORS DONE?
      JNE FLSH         NO, DO IT AGAIN
      AI VDPADD, >FFFD   POINT TO Y BYTE OF HAT
      BL @WRTCL        BLOW FACE OFF SCREEN
      DATA 1, HCO      OFF SCREEN
      C *R0+, *R0+     AND THE FACE
      BL @WRTCL
      DATA 1, HCO      OFF SCREEN
      LWPI MYWS         LOAD MYWS
      LI R10, >1000
      BL @SCORE        GETS 1000 POINTS
      LWPI MYWS2       RESTORE BLWP WS
      AB @H01, @CHAKIL ONE MORE GONE!!
      RTWP             THAT'S ALL FOLKS!!

```

```

*****
* ROUTINE BRANCHED TO FROM MOVCHA *
* THAT DETERMINES IF THE CHASER CAN *
* GO R OR L. BRANCHES BACK TO @MOVCHA*
* AFTER SETTING R1 AS A GOOD/BAD FLAG *
*****

```

```

CLRMOV MOV R7, VDPADD    SET UP VDPADD
      BL @RDCL
      DATA 1, MYWS2+8    Y INTO R4 MSB
      AB @H01, R4        SCREEN STARTS AT >FF
      SRL R4, 11         CL LSB & / BY 8
      SLA R4, 5          x BY 32 = SIT ROW
      INC VDPADD         GOING TO GET X
      BL @RDCL
      DATA 1, MYWS2+12   X INTO R6 MSB
      A R9, R6           R9 WAS PASSED - HAS INCREMENT
      SRL R6, 11         CL LSB & / BY 8
      A R6, R4           = SIT ADDRESS WHERE SPRITE IS
      MOV R4, VDPADD     SET UP VDPADD
      BL @RDCL
      DATA 1, MYWS2+10   CHAR IN R5 MSB
*
      AI VDPADD, 32       POINT DOWN ONE ROW
      BL @RDCL
      DATA 1, MYWS2+11   CHAR IN R5 LSB
      CI R5, >0B0B       FILL CHAR?
      JEQ OK             NO
*

```

```

      JMP NOTOK          GO BACK , NOT OK TO JUMP

```

```

*****
* ROUTINE BRANCHED TO FROM MOVCHA *
* THAT DETERMINES IF THE CHASER CAN *
* MOVE UP/DWN. BRANCHES BACK TO @MOVCHA*
* AFTER SETTING R1 AS A GOOD/BAD FLAG *
*****

```

```

CUDMOV MOV R7, VDPADD    SET UP VDPADD

```

```

BL @RDCL
DATA 1, MYWS2+8      Y INTO R4 MSB
AB @H01, R4          SCREEN STARTS AT >FF
A R9, R4              R9 CONTAINS INCREMENT VALUE
SRL R4, 11           CL LSB & / BY 8
SLA R4, 5             x BY 32 = SIT ROW
INC VDPADD           GOING TO GET X
BL @RDCL
DATA 1, MYWS2+12     X INTO R6 MSB
SRL R6, 11           CL LSB & / BY 8
A R6, R4              = SIT ADDRESS WHERE SPRITE IS
MOV R4, VDPADD       SET UP VDPADD
BL @RDCL
DATA 2, MYWS2+10     CHAR IN R5 MSB
CI R5, >0808         FILL CHARACTER?
JNE NOTOK            NO, SO CAN'T MOVE

*
*
OK      LI R1, 1      SET MOVE FLAG TO OK
        JMP UPRTRN    AND RETURN
NOTOK   CLR R1       SET MOVE FLAG TO NOT OK
*
UPRTRN B @MCCONT     GO BACK
*****
* COINCIDENCE ROUTINE BETWEEN CHASERS *
* AND KANGO. CHASER POSITION IS PASSED*
* IN R5 AND KANGO POSITION IN R6. RE- *
* TURN IS IN R14. *
*****
COINCK  MOV R11, R12  SAVE RETURN
        LI VDPADD, SAL POINT TO KANGOO
        BL @RDCL
        DATA 2, MYWS2+12 R6 NOW HAS KANGO Y, X
        MOV R6, R10    COPY KANGO Y, X
        SRL R6, 8      GOT THE KANGOO Y
        ANDI R10, >00FF AND THE X
        LI R4, 3       LOOP COUNTER
        LI VDPADD, SAL+8 START OF CHASERS
COINCL  BL @RDCL
        DATA 2, MYWS2+2 INTO R1
        MOV R1, R5
        SRL R1, 8      Y INTO LSB OF 1
        CI R1, >00CO   IS THIS CHASER THERE?
        JHE NOTHER    NO, SO SKIP THIS ONE

*
        MOV R5, R9    COPY CHASER POSITION
        SRL R5, 8     CHASER Y INTO LSB OF 5
        ANDI R9, >00FF CHASER X IS IN LSB OF 7

*
        S R6, R5     GET Y DIFFERENCE
        ABS R5       AND MAKE POSITIVE
        S R10, R9    GET X DIFFERENCE
        ABS R9       AND MAKE POSITIVE
*===> KANGO INFORMATION IS STILL PRESERVED
*
        CI R5, 8     4 PIXEL TOLERANCE?
        JH NOTHER    NO COINCIDENCE HERE
        CI R9, 8     4 PIXEL TOLERANCE?
        JH NOTHER    NO COINCIDENCE HERE, EITHER!

*
GOTKAN LWPI MYWS     RESTORE ORIGINAL WORKSPACE
        JMP LOSTK    NAILED TO THE CROSS!

```

```

*
NOTHER AI VDPADD,0          POINT TO NEXT CHASER
      DEC R4                DECREMENT LOOP COUNTER
      JNE COINCL           NOT DONE WITH ALL CHASERS
NCOINK B *R12              NO COINC, SO RETURN TO CALLER
*****
* LOST KANGO ROUTINE. NO RETURN IS *
* NEEDED SINCE IT BRANCHES TO MAINLOOP*
*****
* DELETE CHASERS. RETURN IS IN R12. *
*****
LOSTK LI VDPADD,>308      SAL ADDRESS OF CHASERS
DELCHA LI R6,6           LOOP COUNTER
DCLOOP BL @WRTCL
      DATA 1,HCO
      C *R0+,*R0+        POINT TO NEXT CHASER SPRITE
      DEC R6             DECREMENT LOOP COUNTER
      JNE DCLOOP        NO DONE WITH ALL, YET!!
*****
* DELETE KANGO. RETURN IS IN R12. *
*****
DELKAN LI VDPADD,SAL+2    PATTERN BYTE
      LI R5,>8C00        LEFT PATTERN
      LI R6,>9000        RIGHT PATTERN
      LI R7,30          LOOP COUNTER
DKLOOP BL @WRTCL
      DATA 1,MYWS+10   FACE LEFT
      BL @KILTIM
      DATA 500
      BL @SOUND
      DATA DILSND      AND MAKE SOUND
      BL @WRTCL
      DATA 1,MYWS+12   FACE RIGHT
      BL @KILTIM
      DATA 500
      BL @SOUND
      DATA DIRSND      AND MAKE SOUND
      DEC R7            DONE WITH LOOP?
      JNE DKLOOP
*
      BL @SOUND
      DATA SNDOFF      TURN OFF SOUND GEN
*
      BL @WRTVCL
      DATA SAL,1,HCO
*
      MOVB @HOWMNY,@HOWMNY 2 PLAYERS?
      JNE WHOLST        YES, SO WHO JUST LOST A KANGO?
      B @LKCONT         NO, SO PLAYER 1 JUST LOST IT
*
WHOLST CB @WHICH1,@HO2    DID PLAYER 2 LOSE?
      JEQ TWOLST        YES
*
ONELST SB @HO1,@KANGS1    THERE @OES 1 FOR 1
      JNE OLCONT        BUT HIS GAME IS NOT OVER YET
      BL @WRTVCL        NOW IT IS
      DATA SIT+>143,20,NXTPLB SO TELL HIM
      AI VDPADD,>20
      BL @WRTCL
      DATA 20,DONE1
      AI VDPADD,>20

```

```

BL @WRTCL
DATA 20,NXTPLB
*
ONELL LI R9,5
BL @KILTIM          KEEP IT ON SCREEN
DATA >FFFF
DEC R9
JNE ONELL
JMP OLCNT2          AND GET READY FOR PLAYER 2
*
OLCONT SB @H01,@GAMLV1  KEEP PLAYER 1 AT SAME LEVEL
OLCNT2 BL @WRTVCL
DATA SIT+>143,20,READY2  LET'S GO PLAYER 2
AI VDPADD,>20
BL @WRTCL
DATA 20,NXTPLB        BLANK LINE
AI VDPADD,>20
BL @WRTCL
DATA 20,NEXTPL        PRESS A KEY MSG
LI R10,>00BE          SCADDR FOR 2
MOV R10,@SCADDR
MOVB @H02,@WHICH1    PLAYER 2'S TURN
JMP PLTIMK          AND WASTE SOME TIME
*
TWOLST SB @H01,@KANQS2  2 HAS LOST ONE
JEQ TESTHI          NONE LEFT, SO GAME IS TRULY OVER
SB @H01,@GAMLV2    STAY AT SAME LEVEL
BL @WRTVCL
DATA SIT+>143,20,READY1  LET'S GO 1
AI VDPADD,>20
BL @WRTCL
DATA 20,NXTPLB        BLANK LINE
AI VDPADD,>20
BL @WRTCL
DATA 20,NEXTPL        PRESS A KEY MSG
LI R10,>005E          PLAYER 1 SCORE ADDRESS
MOV R10,@SCADDR
MOVB @H01,@WHICH1
*
PLTIMK CLR @KEYBRD      SET TO BASIC
PLTL1  BL @SCANKY
MOVB @STATUS,R3     CHECK THE STATUS BYTE
JNE PLTL1          DEBOUNCE IS NOT DONE
*
PLTLJ1 MOVB @H01,@KEYBRD  NOW CHECK JOY BUTTON
BL @SCANKY
CB @KEY,@HFF        IS THE BUTTON STILL DOWN?
JNE PLTLJ1          DEBOUNCE IS NOT DONE
*
PLTLJ2 MOVB @H02,@KEYBRD  NOW CHECK JOY BUTTON
BL @SCANKY
CB @KEY,@HFF        SEE IF THE BUTTON IS STILL DOWN
JNE PLTLJ2          DEBOUNCE IS NOT DONE
*
PLTL2  CLR @KEYBRD
CLR @SCNTIM
BL @SCANKY          NOW WAIT FOR A NEW KEY
MOVB @STATUS,R3     KEY PRESSED?
JNE OKTOGO          YES, SO HIT IT!
*
MOVB @H01,@KEYBRD
BL @SCANKY

```

```

MOV B @STATUS, R3
JNE OKTOGO

*

MOV B @H02, @KEYBRD
BL @SCANKY
MOV B @STATUS, R3
JNE OKTOGO

*
NOGOJO JMP PLTL2          NO KEY OR BUTTON HAS BEEN PRESSED
OKTOGO B @NEWLVL        NEXT PLAYER, PLEASE
*****
* TEST FOR HIGH SCORE *
*****
TESTHI LI VDPADD, >59    GET SCORE1
BL @ISITHI              AND TEST FOR HI
LI VDPADD, >B9          GET SCORE2
BL @ISITHI              AND TEST FOR HI
*****
* CHECK FOR REDO OR BACK. *
*****
REDBAK MOV B @DEM, @DEM  ARE WE IN DEMO MODE
JNE RGOON
B @HOPPER              YES, GO BACK TO TITLES
RGOON LI VDPADD, >146    SIT FOR MESSAGE
BL @WRTCL
DATA 14, GAMED        GAME

*

AI VDPADD, >20          SIT FOR MESSAGE
BL @WRTCL
DATA 14, PRESS1       PRESS:
AI VDPADD, >20
BL @WRTCL
DATA 14, PRESS2       REDO OR BACK

*
RBLOOP CLR @SCNTIM      NO SCREEN TIMEOUT
CLR @KEYBRD
BL @SCANKY            GET SOME INPUT
MOV B @KEY, R3
SRL R3, 8             KEY IN R3 LSB
CI R3, >05            '+'
JEQ NOMORE
CI R3, 56             8 (REDO)
JEQ GOREDO
CI R3, 57             9 (BACK)
JEQ GOBACK
CI R3, >0F           BACK
JEQ GOBACK
CI R3, >06           REDO
JNE RBLOOP           WRONG KEY PRESSED

*
GOREDO MOV B @H00, @GAMLV1
MOV B @H00, @GAMLV2
B @GOGAME

*
NOMORE LWP I @PLWS
BLWP @>0000          RETURN TO COLOR BAR

*
GOBACK MOV B @H00, @GAMLV1
MOV B @H00, @GAMLV2
B @HOPPER           START THE GAME OVER

*
LKCONT SB @H01, @GAMLV1 STAY AT SAME LEVEL FOR PLAYER 1

```

```

SB @H01,@KANGS1   AND LOST ONE KANGO
JNE LKCRTN        ALL KANGOS GONE?
B @TESTHI        YES, SO END GAME
LKCRTN B @NEWLVL   NO, SO KEEP GOING
*****
* ROUTINE TO DETERMINE HIGH SCORE *
* RETURN IS IN R15. *
*****
ISITHI MOV R11,R15
          BL @RDCL          SAVE RETURN
          DATA 6,MYWS+8    GET HIGH SCORE
          LI VDPADD,>119
          BL @RDCL
          DATA 6,MYWS+16
          LI R13,MYWS+8     POINT TO REGISTERS
          LI R14,MYWS+16    DITTO, BLUTO
          LI R7,4           LOOP COUNTER
TEST     DEC R7           PASS THROUGH 3 TIMES
          JEQ SAME         DONE, SO THEY MUST BE THE SAME
          C *R13+,*R14+    TESTING TWO BYTES AT A TIME
          JEQ TEST
          JL SAME
NEWHI    BL @WRTCL        REWRITE HIGH SCORE FROM SCORE
          DATA 6,MYWS+8
SAME     B *R15          RETURN TO CALLER
*****
* ADD TO SCORE ROUTINE *
* VALUE TO BE ADDED PASSED IN R10 (>0050=50)
*****
SCORE   MOV R11,R15      SAVE RETURN LINKAGE
          CLR @DIGTHB     DIGTHB EQU 6 (R6)
          CLR @CARYHB     CARYHB EQU 7 (R7)
          MOV @SCADDR,VDPADD WHO'S SCORING NOW?
*       LI VDPADD,>005E   SCREEN ADDR OF LAST DIGIT
RDIGIT  BL @RDCL
          DATA 1,DIGTLB   DIGTLB EQU MYWS+13
          MOV R10,@SAVNUM   SAVE SCORE DIGIT
          ANDI R10,>000F    USE ONLY ONE DIGIT NOW
          A R10,@DIGTHB     ADD THAT DIGIT
          A @CARYHB,@DIGTHB AND ANY CARRY
          CLR @CARYHB       THEN CLEAR THE CARRY
          CB @DIGTLB,@H39   H39 BYTE >39
          JLE WDIGIT       NOT AT AN ASCII '9'
          MOVB @H01,@CARYLB CARYLB EQU MYWS+15
          SB @H0A,@DIGTLB  MAKE THE DIGIT A ZERO
WDIGIT  BL @WRTCL
          DATA 1,DIGTLB   WRITE OUT OUR NEW DIGIT
          MOV @SAVNUM,R10   RESTORE NEXT DIGIT
          SRL R10,4
          DEC VDPADD        POINT TO NEXT SCREEN DIGIT
          MOV @SCADDR,@SAVSAD TEMPORARY LOCATION
          SB @H05,@SAVSAD+1
*       C VDPADD,@SAVSAD  THIS POINTS TO THE FIRST DIGIT
*       CI VDPADD,>0059   SCREEN ADDR OF FIRST DIGIT
          JHE RDIGIT
SCORRT  B *R15          RETURN TO CALLER
*****
* MODULE TO GENERATE A 16 BIT RANDOM *
* NUMBER. PASSED BACK IN R10. RETURN *
* LINKAGE IS IN R11. *
*****
RANDNO  LI R9,28645      MULT FACTOR

```

```

MPY @SEED,R9      SEED EQU >83C0
AI R10,31417     CAUSE SHIFT
MOV R10,@SEED    MAKE NEW SEED
B *R11           AND RETURN

```

```

*****
* MODULE TO GENERATE SOUND. SOUND LIST*
* ADDRESS IS PASSED IN A DATA STATEMENT
* FOLLOWING THE BL. RETURN IS IN R11. *
*****

```

```

SOUND MOV *R11+,@>83CC ADDRESS OF SOUND LIST
      LIM1 0           DISABLE INTERRUPTS
      MOVB @H01,@>83CE TRIGGER SOUND PROCESSING
      SOCB @H01,@>83FD SET VDP RAM FLAG
      LIM1 2           ENABLE INTERRUPTS
      B *R11           AND RETURN

```

```

*****
* THIS ROUTINE IS A SIMULATION OF THE *
* BASIC COMMAND 'CALL HCHAR'. THE *
* PARAMETERS ARE PASSED FOLLOWING THE *
* BL @VCHAR IN DATA STATEMENTS. I. E. : *
* BL @VCHAR *
* DATA >0A0A,>2A05 *
* FORMAT: ROW!COL!!CHAR!!COUNT *
* RETURN LINKAGE IS IN R15 *
*****

```

```

HCHAR MOVB *R11+,R4   START ROW IN 4
HCCC  MOVB *R11+,R5   START COL IN 5
HCC   MOVB *R11+,R6   CHAR VALUE IN 6
HC    MOVB *R11+,R7   COUNT IN 7
H     MOV R11,R15     AND SAVE RETURN LINK

```

```

*
      SRL R4,8        }
      SRL R5,8        } RIGHT JUSTIFY
      SRL R7,8        }
      DEC R4
      DEC R5           OFFSET IS ZERO!
      SLA R4,5        MULTIPLY BY 32 TO GET ROW
      A R5,R4         ADD COL VALUE
*                               R4 CONTAINS SIT START ADDRESS

```

```

HCHARL MOV R4,VDPADD
      BL @WRTCL
      DATA 1,MYWS+12 WRITE ONE CHARACTER
      INC R4           POINT TO NEXT COL
      CI R4,>2FF      GONE OVER COL 32 OF R 24?
      JLE HLCONT     NO, SO CONTINUE
      LI R5,>0300     YES, SO SUBTRACT >300
      S R5,R4         TO POINT TO FIRST ROW

```

```

HLCONT DEC R7
      JNE HCHARL

```

```

*
HCHART B *R15
*****
* THIS ROUTINE IS A SIMULATION OF THE *
* BASIC COMMAND 'CALL VCHAR'. THE *
* PARAMETERS ARE PASSED FOLLOWING THE *
* BL @VCHAR IN DATA STATEMENTS. I. E. : *
* BL @VCHAR *
* DATA >0A0A,>2A05 *
* FORMAT: ROW!COL!!CHAR!!COUNT *
* RETURN LINKAGE IS IN R15 *
*****
VCHAR MOVB *R11+,R4   START ROW IN 4

```

```

VCCC  MOV B *R11+,R5      START COL IN 5
VCC   MOV B *R11+,R6      CHAR VALUE IN 6
VC    MOV B *R11+,R7      COUNT IN 7
V     MDV R11,R15         AND SAVE RETURN LINK
*
      SRL R4,8             }
      SRL R5,8             } RIGHT JUSTIFY
      SRL R7,8             }
      DEC R4
      DEC R5               OFFSET IS ZERO!
      SLA R4,5             MULTIPLY BY 32 TO GET ROW
      A R5,R4              ADD COL VALUE
*
VCHARL MOV R4,VDPADD
      BL @WRTCL
      DATA 1,MYWS+12     WRITE ONE CHARACTER
      AI R4,32            POINT TO NEXT ROW
      CI R4,>2FF          GONE OVER ROW 24?
      JLE VLCCONT        NO, SO CONTINUE
      LI R5,>0300         YES, SO SUBTRACT >300
      S R5,R4             TO POINT TO FIRST ROW
      INC R4              AND MOVE OVER 1 COLUMN
VLCCONT DEC R7
      JNE VCHARL
*
VCHART B *R15
*****
* STANDARD KEYBOARD SCAN ROUTINE *
*****
SCANKY LIM1 0             DISABLE INTS
      LWPI GPLWS          GPLWS EQU >83E0
      BL @SCAN            SCAN EQU >000E
      LWPI MYWS
      LIM1 2              ENABLE INTS
      B *R11              RETURN TO CALLER
*****
* ROUTINE TO CLEAR THE SCREEN. *
* RETURN IS IN R12. *
*****
CLEAR  MOV R11,R12
      CLR R4              START ADDRESS
      LI R1,>2020         SPACE CHAR
CLEARL MOV R4,VDPADD
      BL @WRTCL
      DATA 2,MYWS+2     FROM R1
      INCT R4            INC BY 2
      CI R4,>300         END OF SIT?
      JL CLEARL
      B *R12
*****
* R E A D F R O M V D P ROUTINE *
* THIS ROUTINE ASSUMES THAT THE *
* PARAMETERS ARE PASSED FOLLOWING *
* THE BL @RDVCL STATEMENT. I.E. : *
* BL @RDVCL *
* DATA SAL,2,MYWS+2 *
*****
RDVCL  MOV *R11+,VDPADD  VDPADD EQU 0      (R0)
RDCL   MOV *R11+,RCOUNT  RCOUNT EQU 2      (R2)
RDL    MOV *R11+,RLOC    RLOC EQU 3        (R3)
READ   LIM1 0            DISABLE INTS
      SWPB R0            LOW BYTE FIRST

```



```

MOV B VDPADD, @VDPWA MYWS+1, >8C02
SWPB R0 WASTE SOME TIME
MOV B VDPADD, @VDPWA >8C02
NOP WASTE SOME TIME
R00010 MOV B @VDPWD, *R10C+ VDPWD EQU >8800
DEC RCOUNT
JGT R00010
LIMI 2 ENABLE INTS
B *R11 RETURN TO CALLER
*****
* WRITE TO VDP ROUTINE *
* THIS ROUTINE ASSUMES THAT THE *
* PARAMETERS ARE PASSED FOLLOWING *
* THE BL @WRTVCL STATEMENT. I.E. : *
* BL @WRTVCL *
* DATA SAL, 2, MYWS+2 *
*****
WRTVCL MOV *R11+, VDPADD VDPADD EQU 0 (R3)
WRTCL MOV *R11+, WCOUNT WCOUNT EQU 2 (R2)
WRTL MOV *R11+, WLOC WLOC EQU 3 (R3)
WRITE LIMI 0 DISABLE INTS
SWPB R0 LOW BYTE FIRST
MOV B VDPADD, @VDPWA MYWS+1, >8C02
SWPB R0 RESTORE VDPADD
ORI R0, >4000 SET FLAG TO WRITE
MOV B VDPADD, @VDPWA >8C02
NOP WASTE SOME TIME
W00010 MOV B *WLOC+, @VDPWD VDPWD EQU >8C00
DEC WCOUNT
JGT W00010
LI WCOUNT, >4000 VALUE WE OR'D
S WCOUNT, VDPADD RESTORE ORIGINAL VDPADD
LIMI 2 ENABLE INTS
B *R11 RETURN TO CALLER
*****
* VDP WRITE TO REGISTER ROUTINE. THE*
* PARAMETER IS PASSED DOWN AS DATA *
* USING THIS FORMAT: (LB, THEN HB) *
* VALUE TO WRITE/B/VDP REG. # *
* I.E. BL @VWTR *
* DATA >1187 *
* THIS INSTANCE CHANGES VDP REG. 7 *
* TO >11 (CHANGES BACKGROUND COLOR) *
* RETURN LINKAGE IS R11. *
*****
VWTR MOV *R11+, R1
VWTRG LIMI 0 DISABLE INTS
MOV B R1, @VDPWA VDPWA EQU >8C02
SWPB R1 NOW SET UP HIGH BYTE
MOV B R1, @VDPWA
SWPB R1 RESTORE PARAMETER
LIMI 2 ENABLE INTERRUPTS
B *R11 AND RETURN TO CALLER
END

```