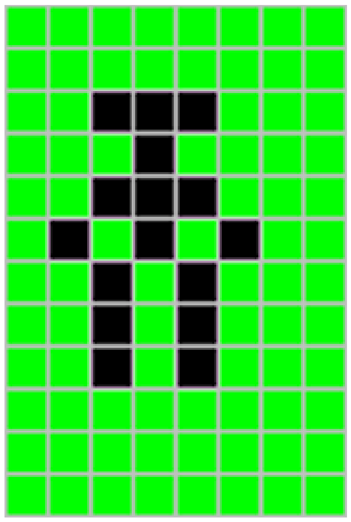
M. David Johnson
http://www.bds-soft.com
info@bds-soft.com



Malky’s Warren: The First Training Quest

Version 1.0.1

by M. David Johnson

2024/05/01

Abstract

A simple PMODE 4 maze game is presented.

At the 2023 CoCoFest, Malky's Warren had been planned to be a 64K CoCo 2 game, but was only working on a CoCo 3 at that time.

For this Version 1.0.1 of 2024, I've fixed several bugs and reorganized the program into a more logical sequence. It's now also running properly on a 64K CoCo 2 as originally planned.

Malky's Warren is intended primarily as a Proof-Of-Concept for the ML Foundation System; including that System's False Disk, Graphics Control, and Fake Text Routines.

In debugging and reorganizing Malky's Warren, no changes were required to those ML Foundation programs.

____

This paper and its associated code are available online at:

http://www.bds-soft.com/cocoPapers.php .

=====

Table of Contents

Introduction

You are an Explorer-In-Training.

______

On February 23, 303 AD, Emperor Diocletian of Rome issued an edict prohibiting Christians from assembling for worship and ordered the destruction of their scriptures, liturgical books, and places of worship across the empire. (Wikipedia).

Many Christians, who were more devoted to Jesus than to the Emperor, hid their scriptures and books in caves; or buried them; or otherwise concealed them rather than destroying them as the edict required.

In the middle of the 20th century, archaeological discoveries at Qumran in Israel, and in the Egyptian desert produced the Dead Sea Scrolls, the Nag Hammadi Library, and other collections of ancient Biblical manuscripts and literature.

______

In the early years of the 21st century, China quietly began cornering the markets for rare-earth minerals and other rare commodities, and began buying up land and businesses around the world; most notably in the United States of America.

On August 25, 2055 AD, the United States Congress proposed the 34th Amendment to the Constitution of the United States which read, “All sovereignty over the United States of America and its territories is hereby ceded to the People's Republic of China (PRC)”. The Amendment was ratified by the States on September 29, 2055 AD.

On January 18, 2056 AD, the United Nations General Assembly issued Resolution 2056-3, ceding sovereignty over the UN to the PRC; and by mid-2056, the entire world was firmly in China’s grip.

______

On February 23, 2063 AD, Emperor Di Jidu Zhe of China issued an edict prohibiting Christians from assembling for worship and ordered the destruction of their scriptures, liturgical books, and places of worship across the entire world.

Many Christians, who were more devoted to Jesus than to the Emperor, hid their scriptures and books in caves; or buried them; or otherwise concealed them rather than destroying them as the edict required.

On October 18, 2077 AD, the world economy suddenly collapsed and civilization was thrown into literal and cultural darkness.

______

On June 8, 2386 AD (June 9, 102 NC [New Calendar]), James Malky was digging out a tree stump on his farm (in what used to be Northwest Colorado) when he discovered a small network of subterranean caves and tunnels. Over the next few months, he explored what soon became known locally as Malky's Warren. In addition to various other artifacts, on November 23, 102 NC, James came upon a bedraggled copy of the Gospel of John.

News of the discovery spread, slowly at first, but then with gathering momentum. By early 116 NC, the search for additional Biblical documents and other artifacts had intensified worldwide; and Malky's Warren was obtained and refitted as a training center for new explorers.

______

As a new Explorer-In-Training, your quest is to enter Malky's Warren, find that Gospel of John, and deliver it to the Warehouse at the Warren's exit. Along the way, you may also find some Provisions to sustain you in your quest.

======

This paper describes the 64K CoCo 2 software which implements Malky's Warren to run on top of the ML Foundation System; including that System's False Disk Routines, Graphics Control Routines, and Fake Text Routines.

Malky's Warren is intended primarily as a Proof-Of-Concept. As such, the Quest is quite simple and easy to traverse. Future Quests won't be that simple (cf. Bippi's Cave: The Second Training Quest).

______

A few General Guidelines:

1. To start the Quest, put the MALKYS.DSK into Drive 0 and enter RUN "MALKYS.BAS".

2. The moment you exit the Warren, the game is over. There's no going back at that point. Be careful not to go East from the Warehouse (marked "W") accidentally.

3. North is up on the screen. Press the " Up-Arrow " to go North. Press the " Right-Arrow " to go East. Press the " Down-Arrow " to go South. Press the " Left-Arrow " to go West.

4. Press the “ T ” Key to Take something and put it in your Bag. Press the “ L ” Key to take something out of your Bag and Leave it in the Current Cell (including the Warehouse Cell).

5. Press the “ B ” Key for a Bag Contents List. Press the “ I ” Key for a Warehouse Inventory List.

6. Press the “G” Key for a New Game. Press the “X” Key to Exit back to CoCo 2 Disk Basic.

______

A Note on Numbers: To keep everything simple to understand, and also neatly lined-up, I frequently refer to numbers as decimal bytes with three full digits, e.g. 004, 027, 229, etc. See Appendix A for conversions between the decimal and hexadecimal representations of bytes. The leading zeros are NOT intended to indicate octal notation. Octal notation is not used anywhere in this paper.

In works of this complexity (at least for me) typos and other errors are bound to sneak in. Please let me know about any you discover so I can note and correct them.

M.D.J. 2024/05/01
info@bds-soft.com

=====

General Methodology

In this paper, the Assembly Language Programming and Listings were prepared using Disk EDTASM+ 01.00.00.

The programs and routines are presented in their general order of relationship to the system as a whole; which, in the case of the game code itself, also happens to be in order of memory location.

The individual programs and routines are fairly well structured internally, and have been significantly modified in the interests of efficiency and effectiveness during the debugging and restructuring process which has been carried out over the past months.

You will note, however, that I have left small sections of unused memory between the various routines in the interests of the debugging and revising processes. Since this program is primarily a "Proof-of-Concept", and since it's working correctly without overloading memory, I saw no need to tighten-up the spacing between the routines.

The code is all fairly well commented and should thus be reasonably easy to follow and understand.

No Testing is recorded in this paper; nor was any significant testing performed other than the simple running of the game at various points during the development cycle. Play the game. Test it for yourself. I'll appreciate any comments or suggestions.

=====

FALSFILE: Reserves the first granule of a disk for Linear Sectors

The BASIC Language program listing:

```
1000 '*****
1010 '*
1020 '* FALSFILE.BAS
1030 '* MDJ 2023/04/19
1040 '*
1050 '* THIS PROGRAM IS BASED
1060 '* UPON THE "FALSINIX.BAS"
1070 '* PROGRAM IN THE FALSE
1070 '* DISK SYSTEM.
1080 '*
1090 '* THIS PROGRAM INITIALIZES
1100 '* A SEMI-FALSE DISK WITH A
1110 '* SINGLE GRANULE (#0) AS A
1120 '* "RESERVED.IMG" FILE.
1130 '*
1140 '* THE 9 SECTORS OF GRANULE 0
1150 '* ARE THEN USED AS LINEAR
1160 '* SECTORS UNDER THE FALSE
1170 '* DISK SYSTEM.
1180 '*
1190 '* THE REMAINING 67 GRANULES
1200 '* ARE AVAILABLE FOR NORMAL
1210 '* PROGRAMS AND FILES.
1220 '*
1210 '* UPON COMPLETION, THE
1220 '* DIRECTORY WILL LOOK
1230 '* LIKE THIS:
1240 '*
1250 '* RESERVED IMG  3 B 1
1260 '*
1270 '* AND WILL PROVIDE
1280 '* THIS RESULT:
1290 '*
1300 '* PRINT FREE(0) --> 67
1310 '*
1320 '*****
1330 '

1500 CLEAR &H1000
1510 '
```

```
1700 PRINT
1710 PRINT "    PUT THE DISK IN DRIVE 0"
1720 PRINT "  ***  ***  WARNING  ***  ***"
1730 PRINT "  *** DISK WILL BE ERASED ***"
1740 PRINT "    PRESS ANY KEY WHEN READY"
1750 A$ = INKEY$
1760 IF A$ = "" GOTO 1750
1770 PRINT
1780 PRINT "WORKING *";
1790 '

2000 'ERASE THE DISK
2010 X$ = "                                "
2020 Z$ = X$+X$+X$+X$
2030 FOR I = 1 TO 128
2040   MID$(Z$,I,1) = CHR$(0)
2050 NEXT I
2060 FOR T = 0 TO 34
2070   FOR S = 1 TO 18
2080     DSKO$ 0,T,S,Z$,Z$
2090   NEXT S
2100   PRINT "*";
2110 NEXT T
2120 '

2200 'GENERATE THE FALSE FAT
2210 F$ = Z$
2220 'SET GRANULE 0
2230 '==> USE ALL 9 SECTORS
2240 MID$(F$,1,1) = CHR$(&HC9)
2250 'SET GRANULES 1 TO 67 = FREE
2260 FOR I = 2 TO 68
2270   MID$(F$,I,1) = CHR$(&HFF)
2280 NEXT I
2290 'PUT TRACK 17, SECTOR 2 TO DISK
2300 DSKO$ 0,17,2,F$,Z$
2310 PRINT "*";
2320 '

2500 'GENERATE THE FALSE DIRECTORY
2510 D$ = Z$
2520 'SET FALSE FILENAME
2531 MID$(D$,1,1) = "R"
2532 MID$(D$,2,1) = "E"
2533 MID$(D$,3,1) = "S"
2534 MID$(D$,4,1) = "E"
```

```
2535 MID$(D$,5,1) = "R"
2536 MID$(D$,6,1) = "V"
2537 MID$(D$,7,1) = "E"
2538 MID$(D$,8,1) = "D"
2539 MID$(D$,9,1) = "I"
2540 MID$(D$,10,1) = "M"
2541 MID$(D$,11,1) = "G"
2600 'SET TYPE =
2610 '  TEXT EDITOR SOURCE
2620 MID$(D$,12,1) = CHR$(3)
2630 'SET FORMAT = BINARY
2640 MID$(D$,13,1) = CHR$(0)
2650 'SET NUMBER OF THE
2660 '  FIRST GRANULE
2670 MID$(D$,14,1) = CHR$(0)
2680 'NUMBER OF BYTES USED
2690 '  IN LAST SECTOR = 256
2700 MID$(D$,15,1) = CHR$(1)
2710 MID$(D$,16,1) = CHR$(0)
2720 'PUT TRACK 17, SECTOR 3 TO DISK
2730 DSKO$ 0,17,3,D$,Z$
2740 PRINT "*";
2750 '

2900 PRINT
2910 PRINT "FALSFILE = DONE"
2920 '

32767 END
```

=====

Sx000001: The Linear Sector Files

There are two 256K Linear Sector Files (SA000001 and SB000001) which were hand-coded to represent the 512 (32x16) character positions of the PMODE 4 Fake Text Screen's representation of Malky's Warren (i.e,. the maze and its reporting fields). This representation is significantly more efficient and economical than a 24-sector graphic representation would require.

There are an additional two 256K Linear Sector Files (SC000001 and SD000001) which are intended to serve as Details and Utilities Sectors for the maze. They are not used in Malky's Warren, but are reserved for future use. At the moment, they are simply dummies (copies of SA000001) and are not presented here.

The Assembly Language text listings:

SA000001:

```
*****
*
* SA000001.ASM
* MDJ 2023/04/07
*
* SCREEN MAKER
*  MAZE    01
*  LEVEL   00
*  SECTION 00
*
* UPPER HALF
*
*****

        ORG     $5500

* LINE 00
LINE00  FCB     32      00
        FCB     32      01
        FCB     32      02
        FCB     32      03
        FCB     32      04
        FCB     32      05
        FCB     32      06
        FCB     32      07
        FCB     32      08
        FCB     105     09
        FCB     98      10
        FCB     103     11
        FCB     107     12
        FCB     103     13
        FCB     107     14
        FCB     103     15
        FCB     107     16
        FCB     103     17
        FCB     107     18
        FCB     103     19
        FCB     107     20
        FCB     103     21
        FCB     107     22
        FCB     103     23
        FCB     107     24
        FCB     103     25
        FCB     107     26
        FCB     103     27
        FCB     107     28
        FCB     103     29
        FCB     99      30
        FCB     32      31

* LINE 01
        FCB     69      00
        FCB     78      01
        FCB     84      02
        FCB     69      03
        FCB     82      04
        FCB     252     05
        FCB     253     06
        FCB     32      07
        FCB     113     08
        FCB     32      09
        FCB     117     10
        FCB     32      11
        FCB     117     12
        FCB     32      13
        FCB     117     14
        FCB     32      15
        FCB     117     16
        FCB     32      17
        FCB     117     18
        FCB     32      19
        FCB     110     20
        FCB     32      21
        FCB     117     22
        FCB     32      23
        FCB     117     24
        FCB     32      25
        FCB     117     26
        FCB     32      27
        FCB     117     28
        FCB     32      29
        FCB     111     30
        FCB     32      31

* LINE 02
        FCB     98      00
        FCB     103     01
        FCB     107     02
        FCB     103     03
        FCB     107     04
        FCB     103     05
        FCB     107     06
        FCB     103     07
        FCB     107     08
        FCB     103     09
        FCB     116     10
        FCB     102     11
        FCB     116     12
        FCB     102     13
        FCB     116     14
        FCB     102     15
        FCB     116     16
        FCB     102     17
        FCB     116     18
        FCB     108     19
        FCB     116     20
        FCB     102     21
        FCB     116     22
        FCB     102     23
        FCB     116     24
        FCB     108     25
        FCB     116     26
        FCB     102     27
        FCB     116     28
        FCB     102     29
        FCB     115     30
        FCB     32      31

* LINE 03
        FCB     111     00
        FCB     32      01
        FCB     117     02
        FCB     32      03
        FCB     117     04
        FCB     32      05
        FCB     117     06
        FCB     32      07
        FCB     117     08
        FCB     32      09
        FCB     117     10
        FCB     32      11
        FCB     117     12
        FCB     32      13
        FCB     117     14
        FCB     32      15
        FCB     117     16
        FCB     32      17
        FCB     117     18
        FCB     32      19
        FCB     110     20
        FCB     32      21
        FCB     117     22
        FCB     32      23
        FCB     117     24
        FCB     32      25
        FCB     117     26
        FCB     32      27
        FCB     117     28
        FCB     32      29
        FCB     111     30
        FCB     32      31

* LINE 04
        FCB     114     00
        FCB     108     01
        FCB     116     02
        FCB     108     03
        FCB     116     04
        FCB     102     05
        FCB     116     06
        FCB     102     07
        FCB     116     08
        FCB     102     09
        FCB     116     10
        FCB     102     11
        FCB     116     12
        FCB     102     13
        FCB     116     14
        FCB     108     15
        FCB     116     16
        FCB     102     17
        FCB     116     18
        FCB     102     19
        FCB     116     20
        FCB     108     21
        FCB     116     22
        FCB     108     23
        FCB     116     24
        FCB     108     25
        FCB     116     26
        FCB     108     27
        FCB     116     28
        FCB     108     29
        FCB     115     30
        FCB     32      31

* LINE 05
        FCB     111     00
        FCB     32      01
        FCB     110     02
        FCB     32      03
        FCB     110     04
        FCB     32      05
        FCB     117     06
        FCB     32      07
        FCB     117     08
        FCB     32      09
        FCB     110     10
        FCB     32      11
        FCB     117     12
        FCB     32      13
        FCB     110     14
        FCB     32      15
        FCB     110     16
        FCB     32      17
        FCB     117     18
        FCB     32      19
        FCB     117     20
        FCB     32      21
        FCB     110     22
        FCB     32      23
        FCB     110     24
        FCB     32      25
        FCB     110     26
        FCB     32      27
        FCB     110     28
        FCB     32      29
        FCB     111     30
        FCB     32      31

* LINE 06
        FCB     114     00
        FCB     108     01
        FCB     116     02
        FCB     108     03
        FCB     116     04
        FCB     108     05
        FCB     116     06
        FCB     102     07
        FCB     116     08
        FCB     108     09
        FCB     116     10
        FCB     108     11
        FCB     116     12
        FCB     108     13
        FCB     116     14
        FCB     108     15
        FCB     116     16
        FCB     108     17
        FCB     116     18
        FCB     102     19
        FCB     116     20
        FCB     102     21
        FCB     116     22
        FCB     102     23
        FCB     116     24
        FCB     102     25
        FCB     116     26
        FCB     108     27
        FCB     116     28
        FCB     108     29
        FCB     115     30
        FCB     32      31

* LINE 07
        FCB     111     00
        FCB     32      01
        FCB     110     02
        FCB     32      03
        FCB     110     04
        FCB     32      05
        FCB     117     06
        FCB     32      07
        FCB     110     08
        FCB     32      09
        FCB     110     10
        FCB     32      11
        FCB     110     12
        FCB     32      13
        FCB     117     14
        FCB     32      15
        FCB     110     16
        FCB     32      17
        FCB     117     18
        FCB     32      19
        FCB     110     20
        FCB     32      21
        FCB     117     22
        FCB     32      23
        FCB     117     24
        FCB     32      25
        FCB     117     26
        FCB     32      27
        FCB     110     28
        FCB     32      29
        FCB     111     30
        FCB     32      31

        END

*****
*
* EOF
*
*****
```

SB000001:

```
*****
*
* SB000001.ASM
* MDJ 2023/04/08
*
* SCREEN MAKER
*  MAZE    01
*  LEVEL   00
*  SECTION 00
*
* LOWER HALF
*
*****
```

```
        ORG     $5600

* LINE 08
        FCB     114     00
        FCB     108     01
        FCB     116     02
        FCB     108     03
        FCB     116     04
        FCB     102     05
        FCB     116     06
        FCB     102     07
        FCB     116     08
        FCB     108     09
        FCB     116     10
        FCB     102     11
        FCB     116     12
        FCB     102     13
        FCB     116     14
        FCB     102     15
        FCB     116     16
        FCB     102     17
        FCB     116     18
        FCB     102     19
        FCB     116     20
        FCB     108     21
        FCB     116     22
        FCB     102     23
        FCB     116     24
        FCB     102     25
        FCB     116     26
        FCB     102     27
        FCB     116     28
        FCB     108     29
        FCB     115     30
        FCB     32      31

* LINE 09
        FCB     111     00
        FCB     32      01
        FCB     110     02
        FCB     32      03
        FCB     117     04
        FCB     32      05
        FCB     117     06
        FCB     32      07
        FCB     117     08
        FCB     32      09
        FCB     117     10
        FCB     32      11
        FCB     110     12
        FCB     32      13
        FCB     110     14
        FCB     32      15
        FCB     117     16
        FCB     32      17
        FCB     117     18
        FCB     32      19
        FCB     117     20
        FCB     32      21
        FCB     110     22
        FCB     32      23
        FCB     117     24
        FCB     32      25
        FCB     117     26
        FCB     32      27
        FCB     117     28
        FCB     32      29
        FCB     111     30
        FCB     32      31

* LINE 10
        FCB     114     00
        FCB     108     01
        FCB     116     02
        FCB     102     03
        FCB     116     04
        FCB     102     05
        FCB     116     06
        FCB     102     07
        FCB     116     08
        FCB     102     09
        FCB     116     10
        FCB     108     11
        FCB     116     12
        FCB     108     13
        FCB     116     14
        FCB     108     15
        FCB     116     16
        FCB     108     17
        FCB     116     18
        FCB     102     19
        FCB     116     20
        FCB     102     21
        FCB     116     22
        FCB     103     23
        FCB     106     24
        FCB     103     25
        FCB     106     26
        FCB     103     27
        FCB     106     28
        FCB     103     29
        FCB     101     30
        FCB     32      31

* LINE 11
        FCB     111     00
        FCB     32      01
        FCB     117     02
        FCB     32      03
        FCB     110     04
        FCB     32      05
        FCB     117     06
        FCB     32      07
        FCB     117     08
        FCB     32      09
        FCB     117     10
        FCB     32      11
        FCB     117     12
        FCB     32      13
        FCB     117     14
        FCB     32      15
        FCB     110     16
        FCB     32      17
        FCB     117     18
        FCB     32      19
        FCB     117     20
        FCB     127     21
        FCB     117     22
        FCB     32      23
        FCB     113     24
        FCB     32      25
        FCB     252     26
        FCB     253     27
        FCB     69      28
        FCB     88      29
        FCB     73      30
        FCB     84      31

* LINE 12
        FCB     100     00
        FCB     103     01
        FCB     106     02
        FCB     103     03
        FCB     106     04
        FCB     103     05
        FCB     106     06
        FCB     103     07
        FCB     106     08
        FCB     103     09
        FCB     106     10
        FCB     103     11
        FCB     106     12
        FCB     103     13
        FCB     106     14
        FCB     103     15
        FCB     106     16
        FCB     103     17
        FCB     106     18
        FCB     103     19
        FCB     106     20
        FCB     103     21
        FCB     101     22
        FCB     105     23
        FCB     32      24
        FCB     32      25
        FCB     32      26
        FCB     32      27
        FCB     32      28
        FCB     32      29
        FCB     32      30
        FCB     32      31

* LINE 13
        FCB     32      00
        FCB     32      01
        FCB     32      02
        FCB     32      03
        FCB     32      04
        FCB     32      05
        FCB     32      06
        FCB     32      07
        FCB     32      08
        FCB     32      09
        FCB     32      10
        FCB     32      11
        FCB     32      12
        FCB     32      13
        FCB     32      14
        FCB     32      15
        FCB     32      16
        FCB     32      17
        FCB     32      18
        FCB     32      19
        FCB     32      20
        FCB     32      21
        FCB     32      22
        FCB     32      23
        FCB     32      24
        FCB     32      25
        FCB     32      26
        FCB     32      27
        FCB     32      28
        FCB     32      29
        FCB     32      30
        FCB     32      31

* LINE 14
        FCB     32      00
        FCB     32      01
        FCB     32      02
        FCB     32      03
        FCB     32      04
        FCB     32      05
        FCB     32      06
        FCB     32      07
        FCB     32      08
        FCB     32      09
        FCB     32      10
        FCB     32      11
        FCB     32      12
        FCB     32      13
        FCB     32      14
        FCB     32      15
        FCB     32      16
        FCB     32      17
        FCB     32      18
        FCB     32      19
        FCB     32      20
        FCB     32      21
        FCB     32      22
        FCB     32      23
        FCB     32      24
        FCB     32      25
        FCB     32      26
        FCB     32      27
        FCB     32      28
        FCB     32      29
        FCB     32      30
        FCB     32      31

* LINE 15
        FCB     83      00
        FCB     84      01
        FCB     82      02
        FCB     69      03
        FCB     78      04
        FCB     71      05
        FCB     84      06
        FCB     72      07
        FCB     32      08
        FCB     61      09
        FCB     32      10
        FCB     48      11
        FCB     48      12
        FCB     48      13
        FCB     48      14
        FCB     48      15
        FCB     32      16
        FCB     32      17
        FCB     32      18
        FCB     83      19
        FCB     67      20
        FCB     79      21
        FCB     82      22
        FCB     69      23
        FCB     32      24
        FCB     61      25
        FCB     32      26
        FCB     48      27
        FCB     48      28
        FCB     48      29
        FCB     48      30
        FCB     48      31

        END

*****
*
* EOF
*
*****
```

SMMAKER: Installs the Linear Sector Files on the first granule

The Assembly Language text listing:

```
*****
*
* SMMAKER.ASM
* MDJ 2023/04/19
*
* SCREEN MAZE MAKER
* ASSEMBLY ROUTINE
*
* SAVES THE SA, SB, SC, SD
* FILE CONTENTS, I.E. THE
* SCREEN INFORMATION BUFFERS,
* TO THE "RESERVED.IMG"
* ON A FALSFILE DISK.
*
*****

FLPUT   EQU     $44F2   PUT BUFFER TO FALSE DISK
LINE00  EQU     $5500   START OF SA FILE
LINE08  EQU     $5600   START OF SB FILE
SCDTLS  EQU     $5700   START OF SB FILE
SCUTLS  EQU     $5800   START OF SB FILE

        ORG     $536F

SMMAKE  PSHS    X,Y

* PUT SA FILE CONTENTS TO
* FALSE DISK SECTOR #0
        LDX  #0
        LDY  #LINE00
        JSR  FLPUT

* PUT SB FILE CONTENTS TO
* FALSE DISK SECTOR #1
        LDX  #1
        LDY  #LINE08
        JSR  FLPUT

* PUT SC FILE CONTENTS TO
* FALSE DISK SECTOR #2
        LDX  #2
        LDY  #SCDTLS
        JSR  FLPUT

* PUT SD FILE CONTENTS TO
* FALSE DISK SECTOR #3
        LDX  #3
        LDY  #SCUTLS
        JSR  FLPUT

        PULS    X,Y

ENDCHK  RTS

        END
```

=====

SMMAKER: Installs the Linear Sector Files on the first granule

The BASIC Control Program listing:

```
1000 '*****
1010 '*
1020 '* SMMAKER.BAS
1030 '* MDJ 2023/04/19
1040 '*
1050 '* SCREEN MAZE MAKER
1060 '* BASIC PROGRAM
1070 '*
1080 '* SAVES THE SA, SB, SC, SD
1090 '* FILE CONTENTS TO
1100 '* THE "RESERVED.IMG"
1110 '* ON A FALSFILE DISK.
1120 '*
1130 '*****
1140 '

2000 'SETUP MEMORY
2010 CLEAR 200, &H4000
2020 '

4000 LOADM "MLBASE.BIN"
4010 LOADM "SMMAKER.BIN"
4020 LOADM "SA000001.BIN"
4030 LOADM "SB000001.BIN"
4040 LOADM "SC000001.BIN"
4050 LOADM "SD000001.BIN"
4060 '

5000 PRINT "PLACE FALSFILE DISK IN DRIVE 0"
5010 PRINT "PRESS ANY KEY WHEN READY >";
5020 A$ = INKEY$
5030 IF A$="" GOTO 5020

6000 EXEC &H536F
6010 '

32767 END
```

=====

SMREADER: Reads the Linear Sectors and displays their bytes for checking

The Assembly Language text listing:

```
*****
*
* SMREADER.ASM
* MDJ 2023/04/10
*
* SCREEN MAZE READER
* ASSEMBLY ROUTINE
*
* GETS THE SCREEN
* INFORMATION BUFFERS
* FROM A FALSE DISK
*
*****

FLGET   EQU     $4533   GET BUFFER FROM FALSE DISK
LINE00  EQU     $5500   START OF LINE00 BUFFER
LINE08  EQU     $5600   START OF LINE08 BUFFER
SCDTLS  EQU     $5700   START OF SCDTLS BUFFER
SCUTLS  EQU     $5800   START OF SCUTLS BUFFER

        ORG     $539C

SMREAD  PSHS    X,Y

* GET LINE00 BUFFER CONTENTS
* FROM FALSE DISK SECTOR #0
        LDX  #0
        LDY  #LINE00
        JSR  FLGET

* GET LINE08 BUFFER CONTENTS
* FROM FALSE DISK SECTOR #1
        LDX  #1
        LDY  #LINE08
        JSR  FLGET

* GET SCDTLS BUFFER CONTENTS
* FROM FALSE DISK SECTOR #2
        LDX  #2
        LDY  #SCDTLS
        JSR  FLGET

* GET SCUTLS BUFFER CONTENTS
* FROM FALSE DISK SECTOR #3
        LDX  #3
        LDY  #SCUTLS
        JSR  FLGET

        PULS    X,Y

ENDCHK  RTS

        END
```

=====

SMREADER: Reads the Linear Sectors and displays their bytes for checking

The BASIC Control Program listing:

```
1000 '*****
1010 '*
1020 '* SMREADER.BAS
1030 '* MDJ 2023/04/19
1040 '*
1050 '* SCREEN MAZE READER
1060 '* BASIC PROGRAM
1070 '*
1080 '* GETS THE CONTENTS OF
1090 '* A FALSE DISK'S
1100 '* SECTORS #0 - #3
1110 '* AND PLACES THE DATA
1120 '* IN FOUR WORKING BUFFERS.
1130 '*
1140 '* IT THEN STEPS THROUGH
1150 '* THE BUFFERS TO ALLOW
1160 '* CHECKING OF THE DATA.
1170 '*
1180 '*****
1190 '

2000 'SETUP MEMORY
2010 CLEAR 200, &H4000
2020 '

4000 LOADM "MLBASE.BIN"
4010 LOADM "SMREADER.BIN"
4040 '

5000 PRINT "PLACE FALSFILE DISK IN DRIVE 0"a
5010 PRINT "PRESS ANY KEY WHEN READY >";
5020 A$ = INKEY$
5030 IF A$="" GOTO 5020

5200 EXEC &H539C
5210 '

5500 FOR Y = 0 TO 7
5510   FOR X = 0 TO 31
5520     Z = (Y * 32) + X
```

```
5530     Z1 = Z + &H5500
5540     C = PEEK(Z1)
5550     PRINT C;
5560   NEXT X
5570 NEXT Y

6000 A$ = INKEY$
6010 IF A$="" GOTO 6000

6500 FOR Y = 8 TO 15
6510   FOR X = 0 TO 31
6520     Z = (Y * 32) + X
6530     Z1 = Z + &H5500
6540     C = PEEK(Z1)
6550     PRINT C;
6560   NEXT X
6570 NEXT Y

7000 A$ = INKEY$
7010 IF A$="" GOTO 7000

7500 FOR Y = 16 TO 23
7510   FOR X = 0 TO 31
7520     Z = (Y * 32) + X
7530     Z1 = Z + &H5500
7540     C = PEEK(Z1)
7550     PRINT C;
7560   NEXT X
7570 NEXT Y

8000 A$ = INKEY$
8010 IF A$="" GOTO 8000

8500 FOR Y = 24 TO 31
8510   FOR X = 0 TO 31
8520     Z = (Y * 32) + X
8530     Z1 = Z + &H5500
8540     C = PEEK(Z1)
8550     PRINT C;
8560   NEXT X
8570 NEXT Y

32767 END
```

=====

SMDISPLY: Reads the Linear Sectors and displays the PMODE 4 Screen they represent

The Assembly Language text listing:

```
*****
*
* SMDISPLY.ASM
* MDJ 2023/04/10
*
* SCREEN MAZE DISPLAY
* ASSEMBLY ROUTINE
*
* DISPLAYS THE MAZE
* ON SCREEN, USING THE
* FAKETEXT 32 X 16
* CHARACTER SET FOR
* PMODE 4
*
* ** START NOTE TO MDJ **
* THIS ROUTINE IS SOMEWHAT
* INEFFICIENT, BUT IT'S
* EASY TO UNDERSTAND AND
* IT'S FAST ENOUGH TO BE
* ACCEPTABLE. IT ALSO USES
* $E400-$E7FF IN HIGH MEMORY.
*    ** BOO! HISS! **
*
* FOR FUTURE WORK: REPLACE
* THIS WITH A MORE EFFICIENT
* SMDRAW ROUTINE - SEE THE
* "FUTUREWORK" FOLDER.
* ** END NOTE TO MDJ **
*
*****

PTFCHR  EQU     $5300   FAKE TEXT ROUTINE
LINE00  EQU     $5500   START OF BUFFERS

        ORG     $53C9

SMDISP  JMP     LBL001
```

```
XCOORD  RMB     1
YCOORD  RMB     1
CHRCOD  RMB     2

LBL001  PSHS    A,B,X,Y,U
        LDY     #LINE00
        LDU     #$E400  TEMP CHRCOD STORE

LBL002  LDB     ,Y+     GET THE CHRCOD
        CLRA            EXTEND IT TO 16-BITS
        STD     ,U++    SAVE CHRCOD TO STORE

        CMPY    #$5700  ARE WE DONE?
        BLO     LBL002  GO IF NO

CHROUT  LDU     #$E400  RESET CHRCOD STORE PTR
        LDA     #$FF    SET FIRST XCOORD TO ROLL
        LDB     #$00    SET FIRST YCOORD TO ZERO

LBL003  INCA            INCREMENT XCOORD
        STA     XCOORD
        STB     YCOORD
        CMPA    #32     END OF THE X LINE?
        BLO     LBL004  GO IF NO
        CLRA            SET XCOORD = 0
        STA     XCOORD
        INCB            INCREMENT YCOORD
        STB     YCOORD
        CMPB    #16     END OF SCREEN?
        BLO     LBL004  GO IF NO
        BRA     LBL005  GO IF YES

LBL004  LDX     ,U++    GET CHRCOD FROM STORE
        STX     CHRCOD

        PSHS    A,B,X   PUT CHRCOD TO SCREEN
        LDA     XCOORD
        LDB     YCOORD
        LDX     CHRCOD
        JSR     PTFCHR
        PULS    A,B,X
        BRA     LBL003  RETURN FOR NEXT CHRCOD

LBL005  PULS    A,B,X,Y,U

ENDCHK  RTS
```

END

=====

SMDISPLY: Reads the Linear Sectors and displays the PMODE 4 Screen they represent

The BASIC Control Program listing:

```
1000 '*****
1010 '*
1020 '* SMDISPLY.BAS
1030 '* MDJ 2023/04/19
1040 '*
1050 '* SCREEN MAZE DISPLAY
1060 '* BASIC PROGRAM
1070 '*
1080 '* DISPLAYS THE MAZE
1090 '* ON SCREEN, USING THE
1100 '* FAKETEXT 32 X 16
1110 '* CHARACTER SET FOR
1120 '* PMODE 4
1130 '*
1140 '*****
1150 '

2000 'SETUP MEMORY
2010 CLEAR 200, &H4000
2020 PCLEAR 4
2030 '

4000 LOADM "MLBASE.BIN"
4010 LOADM "SMREADER.BIN"
4020 LOADM "SMDISPLY.BIN"
4050 '

5000 PRINT "PLACE FALSFILE DISK IN DRIVE 0"
5010 PRINT "PRESS ANY KEY WHEN READY >";
5020 A$ = INKEY$
5030 IF A$="" GOTO 5020
5040 '

7000 EXEC &H539C  'SMREADER
7010 '

9500 'SETUP GRAPHICS
9510 PMODE 4,1
```

```
9520 PCLS 1
9530 SCREEN 1,0
9540 '

9610 EXEC &H53C9  'SMDISPLY
9620 '

9700 'HOLD THE SCREEN
9710 GOTO 9710
9720 '

32767 END
```

=====

MKMLBASE: Combines MLCORE.BIN, FLSYS.BIN, MLGC.BIN, and C3216SET.BIN

This combines the ML Foundation, False Disk, Graphics Control, and Fake Text files all into one combined file for easier use during development. The BASIC Language program listing:

```
1000 '*****
1010 '*
1020 '* MKMLBASE.BAS
1030 '* MDJ 2023/04/07
1040 '*
1050 '* MAKES THE
1060 '* MLBASE.BIN FILE
1070 '* BY COMBINING:
1080 '*   MLCORE.BIN
1090 '*   FLSYS.BIN
1100 '*   MLGC.BIN
1110 '*   C3216SET.BIN
1120 '*
1130 '* SO THAT THE SYSTEM
1140 '* CAN BE LOADED AS A
1150 '* SINGLE ENTITY.
1160 '*
1170 '*****
1180 '

1500 CLEAR 200, &H4000
1510 '

2000 LOADM "MLCORE.BIN"
2010 LOADM "FLSYS.BIN"
2020 LOADM "MLGC.BIN"
2030 LOADM "C3216SET.BIN"
2040 '

3000 SAVEM "MLBASE.BIN", &H4000, &H536E, &H4000
3010 '

32767 END
```

=====

MAKEMLKY: Makes the Production File MALKYS.BIN

This is not actually used until all the Assembly Language files have been completed and assembled. It is placed at this position in the document because, like MKMLBASE, it performs an administrative task rather than a game task. The BASIC Language program listing:

```
0100 '*****
0110 '*
0120 '* MAKEMLKY.BAS
0130 '* MDJ 2024/02/17
0140 '*
0150 '* MAKES THE PRODUCTION
0160 '* FILE MALKYS.BIN
0170 '* FROM THE COLLECTION
0180 '* OF INDIVIDUAL
0190 '* SUBROUTINE AND
0195 '* ASSOCIATED FILES.
0200 '*
0210 '* MZ01.DSK IN DRIVE O
0220 '* CONTENTS:
0230 '* MAKEMLKY.BAS
0240 '* MLBASE.BIN
0250 '* SYSVAR.BIN
0260 '* SIBUFF.BIN
0270 '* DECMAL.BIN
0280 '* RANDOM.BIN
0290 '* MCSCCVT.BIN
0300 '* SMREAD.BIN
0310 '* STAVTR.BIN
0320 '* GTFLOC.BIN
0330 '* PTFCHA.BIN
0340 '* PTFVAL.BIN
0350 '* GTFVAL.BIN
0360 '* PROCHK.BIN
0370 '* PTFBYA.BIN
0380 '* PTFWRA.BIN
0390 '* BCHARK.BIN
0400 '* DCHARK.BIN
0410 '* EASTK.BIN
0420 '* GCHARK.BIN
0430 '* ICHARK.BIN
0440 '* LCHARK.BIN
0450 '* NCHARK.BIN
0460 '* NORTHK.BIN
```

```
0470 '* PCHARK.BIN
0480 '* RCHARK.BIN
0490 '* SOUTHK.BIN
0500 '* TCHARK.BIN
0510 '* UCHARK.BIN
0520 '* WESTK.BIN
0530 '* XCHARK.BIN
0540 '* YCHARK.BIN
0550 '*
0560 '* MZ02.DSK IN DRIVE 1
0570 '* CONTENTS:
0580 '* CLRL13.BIN
0590 '* CLRL14.BIN
0600 '* CLRSTR.BIN
0610 '* CLRSCO.BIN
0620 '* PTFSLS.BIN
0630 '* RPTSTR.BIN
0640 '* RPTSCO.BIN
0650 '* GMOVED.BIN
0660 '* GMOVER.BIN
0670 '* MSG001.BIN
0680 '* MSG002.BIN
0690 '* MSG003.BIN
0700 '* MSG004.BIN
0710 '* MSG005.BIN
0720 '* MSG006.BIN
0730 '* MSG007.BIN
0740 '* MSG008.BIN
0750 '* MSG009.BIN
0760 '* MSG010.BIN
0770 '* MSG011.BIN
0780 '* SMGAME.BIN
0790 '* GMLOOP.BIN
0800 '*
0810 '* MALKYS.DSK IN DRIVE 2
0820 '* CONTENTS:
0830 '* RESERVED.IMG
0840 '* MALKYS.BAS
0850 '*
0860 '* MALKYS.BIN WILL BE ADDED
0870 '* TO MALKYS.DSK WHEN
0880 '* MAKEMLKY.BAS IS RUN
0890 '*
1980 '*****
1990 '

2000 'SETUP MEMORY
```

```
2010 CLEAR 200, &H4000
2020 PCLEAR 4
2030 '

4000 LOADM "MLBASE.BIN"
4010 LOADM "SYSVAR.BIN"
4020 LOADM "SIBUFF.BIN"
4030 LOADM "DECMAL.BIN"
4040 LOADM "RANDOM.BIN"
4050 LOADM "MCSCCVT.BIN"
4060 LOADM "SMREAD.BIN"
4070 LOADM "STAVTR.BIN"
4080 LOADM "GTFLOC.BIN"
4090 LOADM "PTFCHA.BIN"
4100 LOADM "PTFVAL.BIN"
4110 LOADM "GTFVAL.BIN"
4120 LOADM "PROCHK.BIN"
4130 LOADM "PTFBYA.BIN"
4140 LOADM "PTFWRA.BIN"
4150 LOADM "BCHARK.BIN"
4160 LOADM "DCHARK.BIN"
4170 LOADM "EASTK.BIN"
4180 LOADM "GCHARK.BIN"
4190 LOADM "ICHARK.BIN"
4200 LOADM "LCHARK.BIN"
4210 LOADM "NCHARK.BIN"
4220 LOADM "NORTHK.BIN"
4230 LOADM "PCHARK.BIN"
4240 LOADM "RCHARK.BIN"
4250 LOADM "SOUTHK.BIN"
4260 LOADM "TCHARK.BIN"
4270 LOADM "UCHARK.BIN"
4290 LOADM "WESTK.BIN"
4300 LOADM "XCHARK.BIN"
4310 LOADM "YCHARK.BIN"

4320 LOADM "CLRL13.BIN:1"
4330 LOADM "CLRL14.BIN:1"
4340 LOADM "CLRSTR.BIN:1"
4350 LOADM "CLRSCO.BIN:1"
4360 LOADM "PTFSLS.BIN:1"
4370 LOADM "RPTSTR.BIN:1"
4380 LOADM "RPTSCO.BIN:1"
4390 LOADM "GMOVED.BIN:1"
4400 LOADM "GMOVER.BIN:1"
4410 LOADM "MSG001.BIN:1"
4420 LOADM "MSG002.BIN:1"
```

```
4430 LOADM "MSG003.BIN:1"
4440 LOADM "MSG004.BIN:1"
4450 LOADM "MSG005.BIN:1"
4460 LOADM "MSG006.BIN:1"
4470 LOADM "MSG007.BIN:1"
4480 LOADM "MSG008.BIN:1"
4490 LOADM "MSG009.BIN:1"
4500 LOADM "MSG010.BIN:1"
4510 LOADM "MSG011.BIN:1"
4520 LOADM "SMGAME.BIN:1"
4530 LOADM "GMLOOP.BIN:1"
4540 '

5000 SAVEM "MALKYS.BIN:2", &H4000, &H69FF, &H4000
5010 '

32767 END
```

=====

SYSVAR: System Variables

The Assembly Language text listing:

```
                   *****
                   *
                   * SYSVAR.ASM
                   * MDJ 2024/02/13
                   *
                   * SYSTEM VARIABLES
                   *
                   *****

549C                       ORG     $549C

549C               STRNTH  RMB     2       CURRENT STRENGTH
549E               SCORE   RMB     2       CURRENT SCORE

                   * GMOK: 1 = RUNNING; 0 = OVER
54A0               GMOK    RMB     1       GAME OVER FLAG

                   * BAG: &20 = EMPTY; $E0 = GOSPEL OF JOHN
54A1               BAG     RMB     1       BAG CONTENTS

                   * WHSE: &20 = EMPTY; $E0 = GOSPEL OF JOHN
54A2               WHSE    RMB     1       WAREHOUSE CONTENTS

                   * DOCVAL: VALUE OF GOSPEL OF JOHN (21 CHAPTERS)
                   * BASED ON RANDOMLY SELECTED DOCUMENT CONDITION:
                   *         MINT = 210
                   *    EXCELLENT = 189
                   *    VERY GOOD = 168
                   *         GOOD = 147
                   *         FAIR = 126
                   *         POOR = 105
54A3               DOCVAL  RMB     1       SCORE VALUE OF DOCUMENT

                   * PROVAL: NUMBER OF POINTS ADDED TO STRENGTH
                   00440 * RANDOMLY SELECTED BETWEEN 25 AND 75
54A4               00450 PROVAL  RMB     1       PROVISIONS
STRENGTH
                   00460
54A5 12            00470 ENDCHK  NOP
                   00480
          0000     00490         END
```

     =====

SIBUFF: Screen Information Buffers

The first four buffers are loaded from disk, as described in the Sx000001 Chapter. The fifth buffer, the General Utilities Buffer, is uninitialized and is intended to be used as a Screen Information scratchpad. This is for future use: this buffer remains unused in Malky's Warren. The Assembly Language text listing:

```
           *****
           *
           * SIBUFF.ASM
           * MDJ 2024/02/13
           *
           * SCREEN INFORMATION
           * BUFFERS
           *
           *****

5500                ORG     $5500

           * MAZE DESIGN
           * UPPER HALF
           * $5500 - $55FF
5500       LINE00   RMB     256

           * MAZE DESIGN
           * LOWER HALF
           * $5600 - $56FF
5600       LINE08   RMB     256

           * MAZE DESIGN
           * SCREEN DETAILS
           * $5700 - $57FF
5700       SCDTLS   RMB     256

           * MAZE DESIGN
           * SCREEN UTILITIES
           * $5800 - $58FF
5800       SCUTLS   RMB     256

           * GENERAL UTILITY
           * BUFFER; UNINITIALIZED
           * $5900 - $59FF
5900       GENUTL   RMB     256

           * END = $59FF

```

```
        0000        00490              END
```

=====

DECMAL: Get the ASCII Decimal Representation of a 16-bit Unsigned Integer

The Assembly Language text listing:

```
                   00100 *****
                   00110 *
                   00120 * DECMAL.ASM
                   00130 * MDJ 2024/02/13
                   00140 *
                   00150 * GET THE ASCII
                   00160 * REPRESENTATION
                   00170 * OF A DECIMAL
                   00180 * NUMBER BETWEEN
                   00190 * 0 AND 65535
                   00200 *
                   00210 * ENTRY CONDITIONS
                   00220 * D = NUMBER
                   00230 *
                   00240 * EXIT CONDITIONS:
                   00250 * DIGIT4 THROUGH DIGIT0
                   00260 * HOLD THE REPRESENTATION
                   00270 *
                   00280 *****
                   00290
5A00               00300         ORG     $5A00
                   00310
5A00 34   06       00320 DECMAL  PSHS    A,B     SAVE THE NUMBER
5A02 20   05       00330         BRA     LBL001
                   00340
5A04               00350 DIGIT4  RMB     1
5A05               00360 DIGIT3  RMB     1
5A06               00370 DIGIT2  RMB     1
5A07               00380 DIGIT1  RMB     1
5A08               00390 DIGIT0  RMB     1
                   00400
                   00410 * PRELOAD THE DIGITS WITH ASCII "0"
5A09 86   30       00420 LBL001  LDA     #$30    = DECIMAL 48 =
"0"
5A0B B7   5A04     00430         STA     DIGIT4
5A0E B7   5A05     00440         STA     DIGIT3
5A11 B7   5A06     00450         STA     DIGIT2
5A14 B7   5A07     00460         STA     DIGIT1
5A17 B7   5A08     00470         STA     DIGIT0
                   00480
```

```
5A1A 35   06        00490         PULS    A,B      RETRIEVE THE
NUMBER
                    00500
                    00510 * FORM DIGIT 4
5A1C 1083 2710      00520 LBLDG4  CMPD    #10000
5A20 25   08        00530         BLO     LBLDG3
5A22 7C   5A04      00540         INC     DIGIT4
5A25 83   2710      00550         SUBD    #10000
5A28 20   F2        00560         BRA     LBLDG4
                    00570
                    00580 * FORM DIGIT 3
5A2A 1083 03E8      00590 LBLDG3  CMPD    #1000
5A2E 25   08        00600         BLO     LBLDG2
5A30 7C   5A05      00610         INC     DIGIT3
5A33 83   03E8      00620         SUBD    #1000
5A36 20   F2        00630         BRA     LBLDG3
                    00640
                    00650 * FORM DIGIT 2
5A38 1083 0064      00660 LBLDG2  CMPD    #100
5A3C 25   08        00670         BLO     LBLDG1
5A3E 7C   5A06      00680         INC     DIGIT2
5A41 83   0064      00690         SUBD    #100
5A44 20   F2        00700         BRA     LBLDG2
                    00710
                    00720 * FORM DIGIT 1
5A46 1083 000A      00730 LBLDG1  CMPD    #10
5A4A 25   08        00740         BLO     LBLDG0
5A4C 7C   5A07      00750         INC     DIGIT1
5A4F 83   000A      00760         SUBD    #10
5A52 20   F2        00770         BRA     LBLDG1
                    00780
                    00790 * FORM DIGIT 0
5A54 CB   30        00800 LBLDG0  ADDB    #$30
5A56 F7   5A08      00810         STB     DIGIT0
                    00820
5A59 39             00830 ENDCHK  RTS
                    00840
          0000      00850         END
```

=====

RANDOM: Returns a Random Number between 0 and R, where R = 1 to R = 65534

The Assembly Language text listing:

```
                   *****
                   *
                   * RANDOM.ASM
                   * MDJ 2024/02/13
                   *
                   * RETURNS A RANDOM
                   * NUMBER BETWEEN
                   * 0 AND R INCLUSIVE
                   * WHERE R IS BETWEEN
                   * 1 AND 65534 ($FFFE)
                   *
                   * WITH R IN REG X, AND
                   * THE RANDOM NUMBER RETURNED
                   * BY THE ML FOUNDATION'S
                   * RNDU16 IN REGY, THE HIGH
                   * BYTE (REG X) OF MU1616'S
                   * 32-BIT RESULT IS THE
                   * DESIRED RANDOM NUMBER HERE.
                   *
                   * NO CHECKING: THE
                   * USER IS RESPONSIBLE
                   * FOR MAKING SURE THAT
                   * R IS WITHIN RANGE
                   *
                   * ENTRY CONDITIONS:
                   * D = THE R VALUE
                   *
                   * EXIT CONDITIONS:
                   * D = THE RANDOM NUMBER
                   *
                   *****

          429D     MU1616  EQU     $429D   16X16 MULTIPLY
          43E2     RNDU16  EQU     $43E2   MLF'S RNG

5A60                       ORG     $5A60

5A60 34   30       RANDOM  PSHS    X,Y
```

```
                   00480
5A62 C3   0001     00490             ADDD     #1       INCREASE R BY 1
5A65 1F   01       00500             TFR      D,X      MOVE R TO REG X
5A67 BD   43E2     00510             JSR      RNDU16   GET MLF RANDOM
#
5A6A 1F   02       00520             TFR      D,Y      MOVE RANDOM #
TO Y
5A6C BD   429D     00530             JSR      MU1616   GO MULTIPLY
5A6F 1F   10       00540             TFR      X,D      MOVE RESULT TO
D
                   00550
5A71 35   30       00560             PULS     X,Y
5A73 39            00570 ENDCHK      RTS
                   00580
          0000     00590             END
```

=====

MCSCCVT: Maze Coordinates to Screen Coordinates Converter

The Assembly Language text listing:

```
                   *****
                   *
                   * MCSCCVT.ASM
                   * MDJ 2024/02/13
                   *
                   * MAZE COORDINATES TO
                   * SCREEN COORDINATES
                   * CONVERTER
                   *
                   * THE MAZE IS A 15 X 6
                   * CELL STRUCTURE,
                   * SUPERIMPOSED UPON
                   * THE 32 X 16 SCREEN
                   *
                   * FOR CHECKING DOOR
                   * OPENINGS, OBSTRUCTIONS,
                   * ETC., THE MAZE COORDS
                   * NEED TO BE CONVERTED
                   * TO SCREEN COORDINATES.
                   *   SX = (MX * 2) + 1
                   *   SY = (MY * 2) + 1
                   *
                   * ENTRY CONDITIONS:
                   * (MAZE COORDINATES)
                   * A = MX-COORDINATE
                   * B = MY-COORDINATE
                   *
                   * EXIT CONDITIONS:
                   * (SCREEN COORDINATES)
                   * A = SX-COORDINATE
                   * B = SY-COORDINATE
                   *
                   *****

5A80                       ORG     $5A80

5A80 48            MCSCCV  LSLA            MX * 2
5A81 4C                    INCA            + 1

5A82 58                    LSLB            MY * 2
```

```
5A83 5C          00500        INCB           + 1
                 00510
5A84 39          00520 ENDCHK  RTS
          0000   00540        END
```

```
                 00530
```

=====

SMREAD: Gets the Screen Information Buffers from a False Disk

KNOWN BUG: Actually, not from a full False Disk, but rather from the RESERVED.IMG False File granule on the game disk itself, as described in the FALSFILE Chapter. This is just a missed revision of terminology - it doesn’t effect gameplay at all.

The Assembly Language text listing:

```
                   *****
                   *
                   * SMREAD.ASM
                   * MDJ 2024/02/12
                   *
                   * SCREEN MAZE READER
                   * ASSEMBLY ROUTINE
                   * WITH JUMP TO SMGAME
                   *
                   * GETS THE SCREEN
                   * INFORMATION BUFFERS
                   * FROM A FALSE DISK
                   *
                   *****

          4533     FLGET   EQU     $4533   GET BUFFER FROM
FALSE DISK
          5500     LINE00  EQU     $5500   START OF LINE00
BUFFER
          5600     LINE08  EQU     $5600   START OF LINE08
BUFFER
          5700     SCDTLS  EQU     $5700   START OF SCDTLS
BUFFER
          5800     SCUTLS  EQU     $5800   START OF SCUTLS
BUFFER
          67E0     SMGAME  EQU     $67E0

5AA0                       ORG     $5AA0

5AA0 34   30       SMREAD  PSHS    X,Y

                   * GET LINE00 BUFFER CONTENTS
                   * FROM FALSE DISK SECTOR #0
5AA2 8E   0000             LDX  #0
5AA5 108E 5500             LDY  #LINE00
```

```
5AA9 BD   4533     00400          JSR  FLGET
                   00410
                   00420 * GET LINE08 BUFFER CONTENTS
                   00430 * FROM FALSE DISK SECTOR #1
5AAC 8E   0001     00440          LDX  #1
5AAF 108E 5600     00450          LDY  #LINE08
5AB3 BD   4533     00460          JSR  FLGET
                   00470
                   00480 * GET SCDTLS BUFFER CONTENTS
                   00490 * FROM FALSE DISK SECTOR #2
5AB6 8E   0002     00500          LDX  #2
5AB9 108E 5700     00510          LDY  #SCDTLS
5ABD BD   4533     00520          JSR  FLGET
                   00530
                   00540 * GET SCUTLS BUFFER CONTENTS
                   00550 * FROM FALSE DISK SECTOR #3
5AC0 8E   0003     00560          LDX  #3
5AC3 108E 5800     00570          LDY  #SCUTLS
5AC7 BD   4533     00580          JSR  FLGET
                   00590
5ACA 35   30       00600          PULS    X,Y
                   00610
5ACC 7E   67E0     00620          JMP     SMGAME
                   00630
5ACF 12            00640 ENDCHK  NOP
                   00650
          0000     00660          END
```

=====

STAVTR: Sets up the Avatar in its Starting Position

The Assembly Language text listing:

```
                   *****
                   *
                   * STAVTR.ASM
                   * MDJ 2024/02/13
                   *
                   * SETUP AVATAR IN
                   * STARTING POSITION
                   *
                   * STARTING MAZE
                   * COORDINATES
                   *   MX = 4
                   *   MY = 0
                   *
                   * ENTRY CONDITIONS:
                   * NONE
                   *
                   * EXIT CONDITIONS:
                   * NONE
                   *
                   *****

                   * PUT FAKE TEXT ROUTINE
          5300     PTFCHR  EQU     $5300

                   * COORDINATES CONVERTER
          5A80     MCSCCV  EQU     $5A80

5AE0                       ORG     $5AE0

5AE0 34   16       STAVTR  PSHS    A,B,X
5AE2 7E   5AEA             JMP     LBL001

                   * CURRENT MAZE COORDINATES
5AE5               MXC     RMB     1
5AE6               MYC     RMB     1

                   * CURRENT CELL CONTENTS
                   * (UNDER THE AVATAR)
5AE7               CELLCC  RMB     1

```

```
                    00500 * NEW MAZE COORDINATES
                    00510 * (DUMMIES ON STARTUP)
5AE8                00520 MXN     RMB     1
5AE9                00530 MYN     RMB     1
                    00540
                    00550 * SET COORDINATES AND
                    00560 * AND CONTENTS
5AEA 86   20        00570 LBL001  LDA     #32
5AEC B7   5AE7      00580         STA     CELLCC
                    00590
5AEF 86   04        00600         LDA     #4
5AF1 B7   5AE5      00610         STA     MXC
5AF4 B7   5AE8      00620         STA     MXN
                    00630
5AF7 5F             00640         CLRB
5AF8 F7   5AE6      00650         STB     MYC
5AFB F7   5AE9      00660         STB     MYN
                    00670
                    00680 * CONVERT CURRENT
                    00690 * COORDINATES
5AFE BD   5A80      00700         JSR     MCSCCV
                    00710
                    00720 * PLACE AVATAR ON
                    00730 * THE SCREEN, USING ITS
                    00740 * CHARACTER CODE EXTENDED
5B01 8E   0000      00750         LDX     #$0000
5B04 BD   5300      00760         JSR     PTFCHR
                    00770
5B07 35   16        00780         PULS    A,B,X
5B09 39             00790 ENDCHK  RTS
                    00800
          0000      00810         END
```

=====

GTFLOC: Get the Memory Location of the Current Screen (X,Y) Coordinates

The Assembly Language text listing:

```
                   00100 *****
                   00110 *
                   00120 * GTFLOC.ASM
                   00130 * MDJ 2024/02/13
                   00140 *
                   00150 * RETURNS THE ADDRESS
                   00160 * OF THE MEMORY LOCATION
                   00170 * OF THE PMODE 4 SCREEN'S
                   00180 * (X,Y) COORDINATES
                   00190 *
                   00200 * ENTRY CONDITIONS:
                   00210 * A = SX-COORDINATE
                   00220 * B = SY-COORDINATE
                   00230 *
                   00240 * EXIT CONDITIONS:
                   00250 * X = LOCATION ADDRESS
                   00260 *
                   00270 *****
                   00280
          5500     00290 LINE00  EQU     $5500   START OF
BUFFERS
                   00300
5B20               00310         ORG     $5B20
                   00320
5B20 7E   5B25     00330 GTFLOC  JMP     LBL001
                   00340
                   00350 * TEMPORARY 16-BIT
                   00360 * EXTENSION OF SX-COORD
5B23      00       00370 XTEMP1  FCB     $00     HIGH BYTE
5B24               00380 XTEMP2  RMB     1       LOW BYTE
                   00390
5B25 B7   5B24     00400 LBL001  STA     XTEMP2  EXTEND THE SX-
COORD
5B28 4F            00410         CLRA            EXTEND THE SY-
COORD
                   00420
                   00430 * MULTIPLY THE EXTENDED SY-COORDINATE
BY 32
                   00440 * USING THE LSL EQUIVALENT LSLA; ROLB
                   00450 * FIVE TIMES
```

```
5B29 58              00460          LSLB            *2
5B2A 49              00470          ROLA
5B2B 58              00480          LSLB            *4
5B2C 49              00490          ROLA
5B2D 58              00500          LSLB            *8
5B2E 49              00510          ROLA
5B2F 58              00520          LSLB            *16
5B30 49              00530          ROLA
5B31 58              00540          LSLB            *32
5B32 49              00550          ROLA
                     00560
5B33 F3   5B23       00570          ADDD    XTEMP1  ADD EXTENDED
SX-COORD
5B36 C3   5500       00580          ADDD    #LINE00 ADD BUFFERS
START ADDRESS
5B39 1F   01         00590          TFR     D,X     MOVE ADDRESS TO
REG X
                     00600
5B3B 39              00610 ENDCHK  RTS
                     00620
          0000       00630          END
```

=====

PTFCHA: Put a Fake Text Character to the Screen and Advance the Cursor

The Assembly Language text listing:

```
                    00100 *****
                    00110 *
                    00120 * PTFCHA.ASM
                    00130 * MDJ 2024/02/13
                    00140 *
                    00150 * PUT A FAKE TEXT
                    00160 * CHARACTER TO THE
                    00170 * PMODE 4 SCREEN AND
                    00180 * ADVANCE THE POSITION
                    00190 *
                    00200 * ENTRY CONDITIONS:
                    00210 * A = X-COORDINATE (0-31)
                    00220 * B = Y-COORDINATE (0-15)
                    00230 * X = CHARACTER CODE (0-255)
                    00240 *     EXTENDED TO 16-BITS
                    00250 *     I.E. ($0000 - $00FF)
                    00260 *
                    00270 * EXIT CONDITIONS:
                    00280 * A = NEW X-COORDINATE
                    00290 * B = SAME Y-COORDINATE
                    00300 *
                    00310 *****
                    00320
          5300      00330 PTFCHR  EQU     $5300
                    00340
5B40                00350         ORG     $5B40
                    00360
5B40 34   06        00370 PTFCHA  PSHS    A,B     SAVE COORDS TO
STACK
                    00380
5B42 BD   5300      00390         JSR     PTFCHR  PUT CHAR TO
SCREEN
                    00400
5B45 35   06        00410         PULS    A,B     RETRIEVE COORDS
5B47 4C             00420         INCA            POINT TO NEXT
POS
                    00430
5B48 39             00440 ENDCHK  RTS
                    00450
          0000      00460         END
```

PTFVAL: Put a Fake Text Character’s Value to the Screen Information Buffers

The Assembly Language text listing:

```
                   *****
                   *
                   * PTFVAL.ASM
                   * MDJ 2024/02/13
                   *
                   * PUT A FAKE TEXT
                   * CHARACTER'S VALUE
                   * TO THE SCREEN
                   * INFORMATION BUFFERS
                   *
                   * ENTRY CONDITIONS:
                   * A = SX-COORDINATE
                   * B = SY-COORDINATE
                   * X = CHARACTER CODE
                   *     EXTENDED TO 16-BITS
                   *
                   * EXIT CONDITIONS:
                   * NONE
                   *
                   *****

                   * EXTERNAL ROUTINE TO
                   * CONVERT THE (SX,SY)
                   * COORDINATES TO THE
                   * PMODE 4 SCREEN ADDRESS
          5B20     GTFLOC  EQU     $5B20

5B60                       ORG     $5B60

5B60 20   02       PTFVAL  BRA     LBL001

5B62               CTEMP1  RMB     1
5B63               CTEMP2  RMB     1

5B64 BF   5B62     LBL001  STX     CTEMP1 TRUNCATE CHAR
CODE
5B67 BD   5B20             JSR     GTFLOC
5B6A F6   5B63             LDB     CTEMP2 GET TRUNCATED
CODE
5B6D E7   84               STB     ,X
```

```
                   00480
5B6F 39            00490 ENDCHK  RTS
                   00500
          0000     00510         END
```

=====

GTFVAL: Get a Fake Text Character’s Value from the Screen Information Buffers

The Assembly Language text listing:

```
                   *****
                   *
                   * GTFVAL.ASM
                   * MDJ 2024/02/13
                   *
                   * GET A FAKE TEXT
                   * CHARACTER'S VALUE
                   * FROM THE SCREEN
                   * INFORMATION BUFFERS
                   *
                   * ENTRY CONDITIONS:
                   * A = SX-COORDINATE
                   * B = SY-COORDINATE
                   *
                   * EXIT CONDITIONS:
                   * B = CHARACTER CODE
                   *
                   *****

                   * EXTERNAL ROUTINE TO
                   * CONVERT THE (SX,SY)
                   * COORDINATES TO THE
                   * PMODE 4 SCREEN ADDRESS
          5B20     GTFLOC  EQU     $5B20

5B80                       ORG     $5B80

5B80 34   10       GTFVAL  PSHS    X

5B82 BD   5B20             JSR     GTFLOC  X = LOCATION
5B85 E6   84               LDB     ,X      B = VALUE

5B87 35   10               PULS    X
5B89 39            ENDCHK  RTS

          0000             END
```

=====

PROCHK: Provisions Check and Assimilation Subroutine

The Assembly Language text listing:

```
         *****
         *
         * PROCHK.ASM
         * MDJ 2024/02/13
         *
         * PROVISIONS CHECK
         * AND ASSIMILATION
         * SUBROUTINE.
         *
         * THIS SUBROUTINE IS CALLED BY:
         *   EASTK.ASM
         *   WESTK.ASM
         *   NORTHK.ASM
         *   SOUTHK.ASM
         *
         * THIS IS DONE THIS WAY
         * IN ORDER TO ENSURE THE
         * ABOVE FOUR KEY HANDLING
         * ROUTINES DO NOT EXCEED
         * THEIR ALLOTED 256 BYTES.
         *
         *****

         * COORDINATES CONVERTER
5A80     MCSCCV  EQU     $5A80

         * PUT CHARACTER VALUE TO
         * SCREEN INFORMATION BUFFERS
5B60     PTFVAL  EQU     $5B60

         * MAZE COORDINATES
         * AND CONTENTS
         * NOTE: THESE VARIABLES ARE
         *   INTERNAL TO STAVTR.ASM
5AE5     MXC     EQU     $5AE5
5AE6     MYC     EQU     $5AE6
5AE7     CELLCC  EQU     $5AE7

         * STRENGTH AND PROVISIONS
         * NOTE: THESE VARIABLES ARE
```

```
                   *   INTERNAL TO SYSVAR.ASM
          549C     STRNTH  EQU     $549C   STRENGTH VALUE
          54A4     PROVAL  EQU     $54A4   PROVISIONS
VALUE

5BA0                       ORG     $5BA0

                   * CHECK FOR PROVISIONS
5BA0 34   16       PROCHK  PSHS    A,B,X
5BA2 B6   5AE7             LDA     CELLCC  CURRENT CELL
CONTENTS
5BA5 81   85               CMPA    #$85    IS IT
PROVISIONS?
5BA7 26   21               BNE     LBLPC1  GO IF NO
5BA9 86   20               LDA     #$20    BLANK SPACE
5BAB B7   5AE7             STA     CELLCC  TO CURRENT
CONTENTS
5BAE F6   54A4             LDB     PROVAL  PROVISIONS
VALUE
5BB1 4F                    CLRA            EXTEND IT
5BB2 F3   549C             ADDD    STRNTH  ADD IT TO THE
STRENGTH
5BB5 FD   549C             STD     STRNTH  SAVE THE NEW
STRNTH
5BB8 F6   5AE7             LDB     CELLCC  NEW CURRENT
CELL CONTENTS
5BBB 4F                    CLRA            EXTEND IT
5BBC 1F   01               TFR     D,X
5BBE B6   5AE5             LDA     MXC     MX-COORDINATE
5BC1 F6   5AE6             LDB     MYC     MY-COORDINATE
5BC4 BD   5A80             JSR     MCSCCV  CONVERT TO
SX,SY
5BC7 BD   5B60             JSR     PTFVAL  PUT TO SCREEN
BUFFER
5BCA 35   16       LBLPC1  PULS    A,B,X

5BCC 39            ENDCHK  RTS

          0000             END
```

=====

PTFBYA: Put an 8-bit Hexadecimal Number to the Screen and Advance the Cursor

The Assembly Language text listing:

```
                   *****
                   *
                   * PTFBYA.ASM
                   * MDJ 2024/02/12
                   *
                   * PUTS AN 8-BIT NUMBER
                   * TO THE PMODE 4 SCREEN
                   * AS TWO HEXADECIMAL DIGITS
                   * AND ADVANCES THE POSITION
                   *
                   * ENTRY CONDITIONS:
                   * A = X-COORDINATE (0-31)
                   * B = Y-COORDINATE (0-15)
                   * X = 8-BIT NUMBER EXTENDED
                   *     TO 16-BITS (0-255)
                   *
                   * EXIT CONDITIONS:
                   * A = NEW X-COORDINATE
                   * B = SAME Y-COORDINATE
                   *
                   *****

                   * SCRATCHPAD VARIABLES
                   * THE 8-BIT NUMBER
          0076     L0076   EQU     $0076

                   * THE HIGH NIBBLE
          0077     L0077   EQU     $0077

                   * THE LOW NIBBLE
          00F3     L00F3   EQU     $00F3

                   * EXTERNAL ROUTINE
                   * ADDRESS
          5B40     PTFCHA  EQU     $5B40

5BE0                       ORG     $5BE0

5BE0 20   02       PTFBYA  BRA     LBL001

```

```
5BE2              00500 XTEMP   RMB     1
5BE3              00510 YTEMP   RMB     1
                  00520
5BE4 B7   5BE2    00530 LBL001  STA     XTEMP   SAVE THE
COORDINATES
5BE7 F7   5BE3    00540         STB     YTEMP
5BEA 1F   10      00550         TFR     X,D     MOVE THE NUMBER
TO A
5BEC 1F   98      00560         TFR     B,A
                  00570
                  00580 * SAVE THE NUMBER
5BEE 97   76      00590         STA     L0076
                  00600
                  00610 * DIVIDE BY 16
5BF0 44           00620         LSRA
5BF1 44           00630         LSRA
5BF2 44           00640         LSRA
BF3 44           00650         LSRA
                  00660
                  00670 * SAVE THE HIGH NIBBLE
5BF4 97   77      00680         STA L0077
                  00690
                  00700 * MULTIPLY BY 16
5BF6 48           00710         LSLA
5BF7 48           00720         LSLA
5BF8 48           00730         LSLA
5BF9 48           00740         LSLA
                  00750
                  00760 * SAVE TEMP RESULT
5BFA 97   F3      00770         STA     L00F3
                  00780
                  00790 * GET THE NUMBER AGAIN
5BFC 96   76      00800         LDA     L0076
                  00810
                  00820 * SUBTRACT TEMP RESULT
5BFE 90   F3      00830         SUBA    L00F3
                  00840
                  00850 * SAVE LOW NIBBLE
5C00 97   F3      00860         STA     L00F3
                  00870
                  00880 * IS LOW NIBBLE <= 9
5C02 81   09      00890         CMPA    #9
                  00900
                  00910 * GO IF NO
5C04 22   04      00920         BHI     LBL002
                  00930
                  00940 * ADD ZERO OFFSET
```

```
5C06 8B   30        00950         ADDA    #48
5C08 20   02        00960         BRA     LBL003
                    00970
                    00980 * ADD "A" OFFSET
5C0A 8B   37        00990 LBL002  ADDA    #55     (65-10)
                    01000
                    01010 * SAVE LOW NIBBLE CHAR
5C0C 97   F3        01020 LBL003  STA     L00F3
                    01030
                    01040 * GET HIGH NIBBLE
5C0E 96   77        01050         LDA     L0077
                    01060
                    01070 * IS HIGH NIBBLE <= 9
5C10 81   09        01080         CMPA    #9
                    01090
                    01100 * GO IF NO
5C12 22   04        01110         BHI     LBL004
                    01120
                    01130 * ADD ZERO OFFSET
5C14 8B   30        01140         ADDA    #48
5C16 20   02        01150         BRA     LBL005
                    01160
                    01170 * ADD "A" OFFSET
5C18 8B   37        01180 LBL004  ADDA    #55     (65-10)
                    01190
5C1A 1F   89        01200 LBL005  TFR     A,B     EXTEND CHAR TO
X
5C1C 4F             01210         CLRA
5C1D 1F   01        01220         TFR     D,X
5C1F B6   5BE2      01230         LDA     XTEMP   RETRIEVE THE
COORDS
5C22 F6   5BE3      01240         LDB     YTEMP
                    01250
                    01260 * PUT HIGH NIBBLE CHAR
                    01270 * TO THE PMODE 4 SCREEN
5C25 BD   5B40      01280         JSR     PTFCHA  (ALSO ADVANCES
THE POS)
                    01290
5C28 B7   5BE2      01300         STA     XTEMP   SAVE THE
COORDINATES
5C2B F7   5BE3      01310         STB     YTEMP
                    01320
                    01330 * GET LOW NIBBLE CHAR
5C2E 96   F3        01340         LDA     L00F3
                    01350
5C30 1F   89        01360         TFR     A,B     EXTEND CHAR TO
X
```

```
5C32 4F           01370         CLRA
5C33 1F    01     01380         TFR     D,X
5C35 B6    5BE2   01390         LDA     XTEMP   RETRIEVE THE
COORDS
5C38 F6    5BE3   01400         LDB     YTEMP
                  01410
                  01420 * PUT LOW NIBBLE CHAR
                  01430 * TO THE PMODE 4 SCREEN
5C3B BD    5B40   01440         JSR     PTFCHA  (ALSO ADVANCES
THE POS)
                  01450
5C3E 39           01460 ENDCHK  RTS
                  01470
           0000   01480         END
```

=====

PTFWRA: Put a 16-bit Hexadecimal Number to the Screen and Advance the Cursor

The Assembly Language text listing:

```
                    *****
                    *
                    * PTFWRA.ASM
                    * MDJ 2024/02/13
                    *
                    * PUTS A 16-BIT NUMBER
                    * TO THE PMODE 4 SCREEN
                    * AS FOUR HEXADECIMAL DIGITS
                    * AND ADVANCES THE POSITION
                    *
                    * ENTRY CONDITIONS:
                    * A = X-COORDINATE (0-31)
                    * B = Y-COORDINATE (0-15)
                    * X = 16-BIT NUMBER EXTENDED
                    *
                    * EXIT CONDITIONS:
                    * A = NEW X-COORDINATE
                    * B = SAME Y-COORDINATE
                    *
                    *****

                    * EXTERNAL ROUTINE
                    * ADDRESS
          5BE0      PTFBYA  EQU     $5BE0

5C60                        ORG     $5C60

5C60 20   04        PTFWRA  BRA     LBL001

5C62                XTEMP   RMB     1
5C63                YTEMP   RMB     1
5C64                NTEMP   RMB     2

5C66 B7   5C62      LBL001  STA     XTEMP   SAVE THE COORDINATES
5C69 F7   5C63              STB     YTEMP
5C6C BF   5C64              STX     NTEMP   SAVE THE NUMBER

```

```
5C6F FC   5C64              LDD     NTEMP   RETRIEVE THE
NUMBER
5C72 1F   89                TFR     A,B     GET THE HIGH
BYTE TO X
5C74 4F                     CLRA
5C75 1F   01                TFR     D,X
5C77 B6   5C62              LDA     XTEMP   RETRIEVE THE
COORDINATES
5C7A F6   5C63              LDB     YTEMP

                    * PRINT THE HIGH BYTE
5C7D BD   5BE0              JSR     PTFBYA

                    * SAVE THE NEW X-COORDINATE
5C80 B7   5C62              STA     XTEMP

5C83 FC   5C64              LDD     NTEMP   RETRIEVE THE
NUMBER
5C86 4F                     CLRA            GET THE LOW
BYTE TO X
5C87 1F   01                TFR     D,X
5C89 B6   5C62              LDA     XTEMP   RETRIEVE THE
COORDINATES
5C8C F6   5C63              LDB     YTEMP

                    * PRINT THE LOW BYTE
5C8F BD   5BE0              JSR     PTFBYA

5C92 39             ENDCHK  RTS

          0000              END
```

=====

BCHARK: Bag Inventory Key (B-Key) Event Handler

The Assembly Language text listing:

```
                    00100 *****
                    00110 *
                    00120 * BCHARK.ASM
                    00130 * MDJ 2024/02/13
                    00140 *
                    00150 * BAG INVENTORY KEY
                    00160 * (B-KEY)
                    00170 * EVENT HANDLER
                    00180 *
                    00190 *****
                    00200
          54A1      00210 BAG     EQU     $54A1   THE BAG
          64C0      00220 GMOVED  EQU     $64C0   YOU DIED
          6300      00230 CLRL14  EQU     $6300   CLEAR LINE 14
          6660      00240 MSG006  EQU     $6660   "EMPTY BAG"
          66A0      00250 MSG007  EQU     $66A0   "BAG CONTENTS"
                    00260
5CA0                00270         ORG     $5CA0
                    00280
5CA0 34   02        00290 BCHARK  PSHS    A
                    00300
5CA2 BD   6300      00310         JSR     CLRL14  CLEAR LINE 14
                    00320
5CA5 B6   54A1      00330         LDA     BAG     BAG CONTENTS
5CA8 81   E0        00340         CMPA    #$E0    IS IT JOHN?
5CAA 26   05        00350         BNE     LBL001  GO IF NO
5CAC BD   66A0      00360         JSR     MSG007  "BAG CONTENTS"
5CAF 20   03        00370         BRA     LBL002
                    00380
5CB1 BD   6660      00390 LBL001  JSR     MSG006  "EMPTY BAG"
                    00400
5CB4 35   02        00410 LBL002  PULS    A
5CB6 39             00420 ENDCHK  RTS
                    00430
          0000      00440         END
```

=====

DCHARK: Down Key (D-Key) Event Handler

The Assembly Language text listing:

```
                   00100 *****
                   00110 *
                   00120 * DCHARK.ASM
                   00130 * MDJ 2024/02/13
                   00140 *
                   00150 * GO DOWN KEY
                   00160 * (D-KEY)
                   00170 * EVENT HANDLER
                   00180 *
                   00190 * NOT IMPLEMENTED FOR
                   00200 * MALKY'S WARREN
                   00210 *
                   00220 *****
                   00230
5CC0               00240         ORG     $5CC0
                   00250
5CC0 39            00260 DCHARK  RTS
                   00270
          0000     00280         END
```

=====

EASTK: East Key (Right Arrow) Event Handler

The Assembly Language text listing:

```
             *****
             *
             * EASTK.ASM
             * MDJ 2024/02/13
             *
             * EAST KEY
             * (RIGHT ARROW)
             * EVENT HANDLER
             *
             *****

             * PUT FAKE TEXT ROUTINE
5300         PTFCHR  EQU     $5300

             * COORDINATES CONVERTER
5A80         MCSCCV  EQU     $5A80

             * GET CHARACTER VALUE FROM
             * SCREEN INFORMATION BUFFERS
5B80         GTFVAL  EQU     $5B80

             * MAZE COORDINATES
             * AND CONTENTS
             * NOTE: THESE VARIABLES ARE
             *   INTERNAL TO STAVTR.ASM
5AE5         MXC     EQU     $5AE5
5AE6         MYC     EQU     $5AE6
5AE7         CELLCC  EQU     $5AE7
5AE8         MXN     EQU     $5AE8
5AE9         MYN     EQU     $5AE9

             * VERTICAL DOOR CODE
0075         VRTDOR  EQU     $75

             * GAME OVER ROUTINES
6500         GMOVER  EQU     $6500
64C0         GMOVED  EQU     $64C0

             * STRENGTH REPORTING
549C         STRNTH  EQU     $549C
```

```
          6400      00500 RPTSTR  EQU     $6400
                    00510
                    00520 * SCORE REPORTING
          549E      00530 SCORE   EQU     $549E
          6460      00540 RPTSCO  EQU     $6460
                    00550
          5BA0      00560 PROCHK  EQU     $5BA0   PROVISIONS
CHECK
                    00570
                    00580 * MESSAGE ROUTINES
          62C0      00590 CLRL13  EQU     $62C0
          6300      00600 CLRL14  EQU     $6300
          6540      00610 MSG001  EQU     $6540
                    00620
5CE0                00630         ORG     $5CE0
                    00640
5CE0 34   16        00650 EASTK   PSHS    A,B,X
                    00660
5CE2 BD   62C0      00670         JSR     CLRL13  CLEAR LINE 13
5CE5 BD   6300      00680         JSR     CLRL14  CLEAR LINE 14
                    00690
                    00700 * CHECK FOR LEGAL MOVE
5CE8 B6   5AE5      00710         LDA     MXC     MX-COORDINATE
5CEB F6   5AE6      00720         LDB     MYC     MY-COORDINATE
5CEE BD   5A80      00730         JSR     MCSCCV  CONVERT TO
SX,SY
5CF1 4C             00740         INCA            POINT TO NEXT
SX
5CF2 BD   5B80      00750         JSR     GTFVAL  GET THE FAKE
CHAR
5CF5 C1   75        00760         CMPB    #VRTDOR IS IT AN
OPENING
5CF7 27   2C        00770         BEQ     LBL002  GO IF YES
5CF9 BD   6540      00780         JSR     MSG001  DISPLAY ERROR
MESSAGE
                    00790
                    00800 * ADJUST STRENGTH VARIABLE
5CFC 34   06        00810         PSHS    A,B     STRENGTH EFFECT
5CFE FC   549C      00820         LDD     STRNTH
5D01 1083 0002      00830         CMPD    #2      IS IT AT LIMIT
5D05 22   10        00840         BHI     LBL001  GO IF NO
5D07 CC   0000      00850         LDD     #0
5D0A FD   549C      00860         STD     STRNTH
5D0D 35   06        00870         PULS    A,B
5D0F BD   6400      00880         JSR     RPTSTR  REPORT CURRENT
STRENGTH
5D12 35   16        00890         PULS    A,B,X
```

```
5D14 7E   64C0     00900          JMP     GMOVED  YOU DIED
                   00910
5D17 83   0002     00920 LBL001   SUBD    #2
5D1A FD   549C     00930          STD     STRNTH
5D1D 35   06       00940          PULS    A,B
5D1F BD   6400     00950          JSR     RPTSTR  REPORT CURRENT
STRENGTH
5D22 16   009A     00960          LBRA    LBL004  GO (IT'S NOT AN
OPENING)
                   00970
                   00980 * REVEAL THE CURRENT CELL
                   00990 * CONTENTS ON THE SCREEN.
                   01000 * (FROM UNDER THE AVATAR)
                   01010 * TESTING = USE SPACE
5D25 F6   5AE7     01020 LBL002   LDB     CELLCC  CONTENTS
5D28 4F            01030          CLRA            EXTEND IT
5D29 1F   01       01040          TFR     D,X
5D2B B6   5AE5     01050          LDA     MXC     MX-COORDINATE
5D2E F6   5AE6     01060          LDB     MYC     MY-COORDINATE
5D31 BD   5A80     01070          JSR     MCSCCV  CONVERT TO
SX,SY
5D34 BD   5300     01080          JSR     PTFCHR  PUT TO SCREEN
                   01090
                   01100 * GO ONE MAZE CELL EAST
5D37 B6   5AE5     01110          LDA     MXC     CURRENT MX
5D3A 4C            01120          INCA
5D3B B7   5AE8     01130          STA     MXN     NEW MX
5D3E F6   5AE6     01140          LDB     MYC     CURRENT MY
5D41 F7   5AE9     01150          STB     MYN     NEW MY
                   01160
                   01170 * SAVE NEW CELL SCREEN CONTENTS
5D44 BD   5A80     01180          JSR     MCSCCV  CONVERT TO
SX,SY
5D47 BD   5B80     01190          JSR     GTFVAL  GET CHAR VALUE
5D4A F7   5AE7     01200          STB     CELLCC  SAVE AS
CONTENTS
                   01210
                   01220 * PUT AVATAR TO NEW SCREEN LOCATION
5D4D B6   5AE8     01230          LDA     MXN     MX-COORDINATE
5D50 F6   5AE9     01240          LDB     MYN     MY-COORDINATE
5D53 BD   5A80     01250          JSR     MCSCCV  CONVERT TO
SX,SY
5D56 8E   0000     01260          LDX     #$0000  AVATAR CODE
EXTENDED
5D59 BD   5300     01270          JSR     PTFCHR  PUT TO SCREEN
                   01280
                   01290 * MAKE THE NEW COORDINATES CURRENT
```

```
5D5C B6   5AE8     01300          LDA     MXN     NEW MX
5D5F B7   5AE5     01310          STA     MXC     CURRENT MX
5D62 F6   5AE9     01320          LDB     MYN     NEW MY
5D65 F7   5AE6     01330          STB     MYC     CURRENT MY
                   01340
                   01350 * GO CHECK FOR PROVISIONS
5D68 BD   5BA0     01360          JSR     PROCHK
                   01370
                   01380 * ADJUST STRENGTH VARIABLE
5D6B 34   06       01390          PSHS    A,B     STRENGTH EFFECT
5D6D FC   549C     01400          LDD     STRNTH
5D70 1083 0001     01410          CMPD    #1      IS IT AT LIMIT
5D74 22   10       01420          BHI     LBL003  GO IF NO
5D76 CC   0000     01430          LDD     #0
5D79 FD   549C     01440          STD     STRNTH
5D7C 35   06       01450          PULS    A,B
5D7E BD   6400     01460          JSR     RPTSTR  REPORT CURRENT
STRENGTH
5D81 35   16       01470          PULS    A,B,X
5D83 7E   64C0     01480          JMP     GMOVED  YOU DIED
                   01490
5D86 83   0001     01500 LBL003   SUBD    #1
5D89 FD   549C     01510          STD     STRNTH
5D8C 35   06       01520          PULS    A,B
5D8E BD   6400     01530          JSR     RPTSTR  REPORT CURRENT
STRENGTH
                   01540
                   01550 * CHECK FOR QUEST COMPLETION
                   01560 * (CAN ONLY BE ACCOMPLISHED BY AN
                   01570 *  EAST MOVE INTO CELL (MX=11, MY=5)
5D91 34   06       01580          PSHS    A,B
5D93 B6   5AE5     01590          LDA     MXC
5D96 81   0B       01600          CMPA    #11
5D98 26   23       01610          BNE     LBLSC3
5D9A F6   5AE6     01620          LDB     MYC
5D9D C1   05       01630          CMPB    #5
5D9F 26   1C       01640          BNE     LBLSC3
                   01650
                   01660 * ADJUST STRENGTH AND SCORE
                   01670 * TEMPORARILY SKIP LIMIT CHECK
5DA1 FC   549E     01680          LDD     SCORE
5DA4 F3   549C     01690          ADDD    STRNTH
5DA7 FD   549E     01700          STD     SCORE
5DAA CC   0000     01710          LDD     #0
5DAD FD   549C     01720          STD     STRNTH
5DB0 BD   6400     01730          JSR     RPTSTR  REPORT CURRENT
STRENGTH
```

```
5DB3 BD   6460     01740          JSR     RPTSCO  REPORT CURRENT
SCORE
5DB6 35   06       01750          PULS    A,B
5DB8 35   16       01760          PULS    A,B,X
5DBA 7E   6500     01770          JMP     GMOVER  QUEST IS
COMPLETE
                   01780
5DBD 35   06       01790 LBLSC3   PULS    A,B
                   01800
5DBF 35   16       01810 LBL004   PULS    A,B,X
5DC1 39            01820 ENDCHK   RTS
                   01830
          0000     01840          END
```

=====

GCHARK: New Game Key (G-Key) Event Handler

The Assembly Language text listing:

```
                   *****
                   *
                   * GCHARK.ASM
                   * MDJ 2024/02/13
                   *
                   * NEW GAME KEY
                   * (G-KEY)
                   * EVENT HANDLER
                   *
                   * CHECKS FOR CONFIRMATION
                   * THEN STARTS A NEW GAME
                   * IF CONFIRMED.
                   *
                   * RETURNS WITH NO ACTION
                   * IF NOT CONFIRMED
                   *
                   *****

                   * MLF POLCAT
          4142     POLCAT  EQU     $4142

                   * NEW GAME ADDRESS
          5AA0     SMREAD  EQU     $5AA0

                   * CONFIRM MESSAGE EQUATES
          62C0     CLRL13  EQU     $62C0   CLEAR LINE 13
          6300     CLRL14  EQU     $6300   CLEAR LINE 14
          67A0     MSG011  EQU     $67A0   "NEW GAME
CONFIRM"

5DE0                       ORG     $5DE0

5DE0 34   02       GCHARK  PSHS    A

5DE2 BD   6300             JSR     CLRL14  GO CLEAR LINE
14
5DE5 BD   67A0             JSR     MSG011  CONFIRM?

                   * GET A KEYPRESS
5DE8 BD   4142     LBL001  JSR     POLCAT
```

```
5DEB 27   FB                BEQ     LBL001

                   * YCHARK (Y-KEY)
5DED 81   59                CMPA    #$59
5DEF 26   03                BNE     LBL002
5DF1 7E   5AA0              JMP     SMREAD  GO START NEW
GAME

                   * NCHARK (N-KEY)
5DF4 81   4E       LBL002   CMPA    #$4E
5DF6 26   08                BNE     LBL003
5DF8 BD   62C0              JSR     CLRL13  GO CLEAR LINE
13
5DFB BD   6300              JSR     CLRL14  GO CLEAR LINE
14
5DFE 20   03                BRA     LBL004  GO DO RTS

                   * ANY OTHER KEYPRESS
5E00 16   FFE5     LBL003   LBRA    LBL001

5E03 35   02       LBL004   PULS    A
5E05 39            ENDCHK   RTS

          0000              END
```

=====

ICHARK: Warehouse Inventory Key (I-Key) Event Handler

The Assembly Language text listing:

```
                  00100 *****
                  00110 *
                  00120 * ICHARK.ASM
                  00130 * MDJ 2024/02/13
                  00140 *
                  00150 * WAREHOUSE INVENTORY KEY
                  00160 * (I-KEY)
                  00170 * EVENT HANDLER
                  00180 *
                  00190 *****
                  00200
          54A2    00210 WHSE    EQU     $54A2   THE WAREHOUSE
          64C0    00220 GMOVED  EQU     $64C0   YOU DIED
          6300    00230 CLRL14  EQU     $6300   CLEAR LINE 14
          66E0    00240 MSG008  EQU     $66E0   "EMPTY WHSE"
          6720    00250 MSG009  EQU     $6720   "WHSE
INVENTORY"
                  00260
5E20              00270         ORG     $5E20
                  00280
5E20 34   02      00290 ICHARK  PSHS    A
                  00300
5E22 BD   6300    00310         JSR     CLRL14  CLEAR LINE 14
                  00320
5E25 B6   54A2    00330         LDA     WHSE    WHSE CONTENTS
5E28 81   E0      00340         CMPA    #$E0    IS IT JOHN?
5E2A 26   05      00350         BNE     LBL001  GO IF NO
5E2C BD   6720    00360         JSR     MSG009  "WHSE
INVENTORY"
5E2F 20   03      00370         BRA     LBL002
                  00380
5E31 BD   66E0    00390 LBL001  JSR     MSG008  "EMPTY WHSE"
                  00400
5E34 35   02      00410 LBL002  PULS    A
5E36 39           00420 ENDCHK  RTS
                  00430
          0000    00440         END
```

=====

LCHARK: Leave Key (L-Key) Event Handler

The Assembly Language text listing:

```
         *****
         *
         * LCHARK.ASM
         * MDJ 2024/02/12
         *
         * LEAVE KEY
         * (L-KEY)
         * (EMPTY CONTENTS OF BAG INTO CELL)
         * EVENT HANDLER
         *
         *****

54A1     BAG     EQU     $54A1   THE BAG
54A2     WHSE    EQU     $54A2   THE WAREHOUSE
54A3     DOCVAL  EQU     $54A3   DOCUMENT VALUE
64C0     GMOVED  EQU     $64C0   YOU DIED
6300     CLRL14  EQU     $6300   CLEAR LINE 14
6640     MSG005  EQU     $6640   "NO ROOM"
6660     MSG006  EQU     $6660   "EMPTY BAG"

         * PUT FAKE TEXT ROUTINE
5300     PTFCHR  EQU     $5300

         * COORDINATES CONVERTER
5A80     MCSCCV  EQU     $5A80

         * PUT CHARACTER VALUE TO
         * SCREEN INFORMATION BUFFERS
5B60     PTFVAL  EQU     $5B60

         * MAZE COORDINATES
         * AND CONTENTS
5AE5     MXC     EQU     $5AE5
5AE6     MYC     EQU     $5AE6
5AE7     CELLCC  EQU     $5AE7

         * STRENGTH REPORTING
549C     STRNTH  EQU     $549C
6400     RPTSTR  EQU     $6400

```

```
                  * SCORE REPORTING
          549E    SCORE   EQU     $549E
          6460    RPTSCO  EQU     $6460

5E40                      ORG     $5E40

5E40 34   16      LCHARK  PSHS    A,B,X

5E42 BD   6300            JSR     CLRL14  CLEAR LINE 14

                  * CHECK BAG FOR GOSPEL OF JOHN
5E45 B6   54A1            LDA     BAG     GET BAG
CONTENTS
5E48 81   E0              CMPA    #$E0    IS IT JOHN?
5E4A 26   3F              BNE     LBLBC2  GO IF NO
5E4C F6   5AE7            LDB     CELLCC  CURRENT CELL
CONTENTS
5E4F C1   7F              CMPB    #$7F    IS IT THE
WAREHOUSE?
5E51 27   1F              BEQ     LBLBC1  GO IF YES
5E53 C1   20              CMPB    #$20    IS IT A BLANK
SPACE?
5E55 26   39              BNE     LBLBC3  GO IF NO

                  * PUT BAG CONTENTS TO CELL
5E57 86   20              LDA     #$20    EMPTY THE BAG
5E59 B7   54A1            STA     BAG
5E5C C6   E0              LDB     #$E0    PUT JOHN IN THE
CELL
5E5E F7   5AE7            STB     CELLCC
5E61 4F                   CLRA            EXTEND IT
5E62 1F   01              TFR     D,X
5E64 B6   5AE5            LDA     MXC     MX-COORDINATE
5E67 F6   5AE6            LDB     MYC     MY-COORDINATE
5E6A BD   5A80            JSR     MCSCCV  CONVERT TO
SX,SY
5E6D BD   5B60            JSR     PTFVAL  PUT TO SCREEN
BUFFER
5E70 20   49              BRA     LBL002

                  * PUT BAG CONTENTS TO WAREHOUSE
5E72 86   20      LBLBC1  LDA     #$20    EMPTY THE BAG
5E74 B7   54A1            STA     BAG
5E77 86   E0              LDA     #$E0    PUT JOHN IN
WHSE
5E79 B7   54A2            STA     WHSE
```

```
5E7C F6   54A3      00880         LDB     DOCVAL  GET THE
DOCUMENT VALUE
5E7F 4F             00890         CLRA            EXTEND IT
5E80 F3   549E      00900         ADDD    SCORE   ADD IT TO THE
SCORE
5E83 FD   549E      00910         STD     SCORE   SAVE THE NEW
SCORE
5E86 BD   6460      00920         JSR     RPTSCO  REPORT THE NEW
SCORE
5E89 20   30        00930         BRA     LBL002
                    00940
5E8B BD   6660      00950 LBLBC2  JSR     MSG006  "BAG EMPTY"
5E8E 20   03        00960         BRA     LBL000
5E90 BD   6640      00970 LBLBC3  JSR     MSG005  "NO ROOM"
                    00980
                    00990 * FAILED ACTION: ADJUST STRENGTH
VARIABLE
5E93 34   06        01000 LBL000  PSHS    A,B     STRENGTH EFFECT
5E95 FC   549C      01010         LDD     STRNTH
5E98 1083 0002      01020         CMPD    #2      IS IT AT LIMIT
5E9C 22   10        01030         BHI     LBL001  GO IF NO
5E9E CC   0000      01040         LDD     #0
5EA1 FD   549C      01050         STD     STRNTH
5EA4 35   06        01060         PULS    A,B
5EA6 BD   6400      01070         JSR     RPTSTR  REPORT CURRENT
STRENGTH
5EA9 35   16        01080         PULS    A,B,X
5EAB 7E   64C0      01090         JMP     GMOVED  YOU DIED
                    01100
5EAE 83   0002      01110 LBL001  SUBD    #2
5EB1 FD   549C      01120         STD     STRNTH
5EB4 35   06        01130         PULS    A,B
5EB6 BD   6400      01140         JSR     RPTSTR  REPORT CURRENT
STRENGTH
5EB9 20   26        01150         BRA     LBLBC4
                    01160
                    01170 * COMPLETED ACTION: ADJUST STRENGTH
VARIABLE
5EBB 34   06        01180 LBL002  PSHS    A,B     STRENGTH EFFECT
5EBD FC   549C      01190         LDD     STRNTH
5EC0 1083 0001      01200         CMPD    #1      IS IT AT LIMIT
5EC4 22   10        01210         BHI     LBL003  GO IF NO
5EC6 CC   0000      01220         LDD     #0
5EC9 FD   549C      01230         STD     STRNTH
5ECC 35   06        01240         PULS    A,B
5ECE BD   6400      01250         JSR     RPTSTR  REPORT CURRENT
STRENGTH
```

```
5ED1 35   16        01260          PULS    A,B,X
5ED3 7E   64C0      01270          JMP     GMOVED  YOU DIED
                    01280
5ED6 83   0001      01290 LBL003   SUBD    #1
5ED9 FD   549C      01300          STD     STRNTH
5EDC 35   06        01310          PULS    A,B
5EDE BD   6400      01320          JSR     RPTSTR  REPORT CURRENT
STRENGTH
                    01330
5EE1 35   16        01340 LBLBC4   PULS    A,B,X
5EE3 39             01350 ENDCHK   RTS
                    01360
          0000      01370          END
```

=====

NCHARK: “No” (Do Not Confirm) Key (N-Key) Event Handler

The Assembly Language text listing:

```
                   00100 *****
                   00110 *
                   00120 * NCHARK.ASM
                   00130 * MDJ 2024/02/13
                   00140 *
                   00150 * NOT CONFIRMED KEY
                   00160 * (N-KEY)
                   00170 * EVENT HANDLER
                   00180 *
                   00190 * THIS IS AN UNUSED DUMMY
                   00200 * ROUTINE - FOR POTENTIAL
                   00210 * FUTURE USE ONLY
                   00220 *
                   00230 * WOULD DO A SIMPLE RETURN
                   00240 * TO CALLER TO VERIFY
                   00250 * THAT THE PROPOSED ACTION
                   00260 * IS NOT CONFIRMED.
                   00270 *
                   00280 * NO ACTION IF N-KEY IS
                   00290 * PRESSED IN GMLOOP.
                   00300 *
                   00310 *****
                   00320
5F00               00330         ORG     $5F00
                   00340
5F00 39            00350 NCHARK  RTS
                   00360
          0000     00370         END
```

=====

NORTHK: North Key (Up Arrow) Event Handler

The Assembly Language text listing:

```
                00100 *****
                00110 *
                00120 * NORTHK.ASM
                00130 * MDJ 2024/02/13
                00140 *
                00150 * NORTH KEY
                00160 * (UP ARROW)
                00170 * EVENT HANDLER
                00180 *
                00190 *****
                00200
                00210 * PUT FAKE TEXT ROUTINE
5300            00220 PTFCHR  EQU     $5300
                00230
                00240 * COORDINATES CONVERTER
5A80            00250 MCSCCV  EQU     $5A80
                00260
                00270 * GET CHARACTER VALUE FROM
                00280 * SCREEN INFORMATION BUFFERS
5B80            00290 GTFVAL  EQU     $5B80
                00300
                00310 * MAZE COORDINATES
                00320 * AND CONTENTS
                00330 * NOTE: THESE VARIABLES ARE
                00340 *   INTERNAL TO STAVTR.ASM
5AE5            00350 MXC     EQU     $5AE5
5AE6            00360 MYC     EQU     $5AE6
5AE7            00370 CELLCC  EQU     $5AE7
5AE8            00380 MXN     EQU     $5AE8
5AE9            00390 MYN     EQU     $5AE9
                00400
                00410 * HORIZONTAL DOOR CODE
006C            00420 HORDOR  EQU     $6C
                00430
                00440 * GAME OVER ROUTINES
6500            00450 GMOVER  EQU     $6500
64C0            00460 GMOVED  EQU     $64C0
                00470
                00480 * STRENGTH REPORTING
549C            00490 STRNTH  EQU     $549C
```

```
          6400      RPTSTR  EQU     $6400

                    * SCORE REPORTING
          549E      SCORE   EQU     $549E
          6460      RPTSCO  EQU     $6460

          5BA0      PROCHK  EQU     $5BA0   PROVISIONS
CHECK

                    * MESSAGE ROUTINES
          62C0      CLRL13  EQU     $62C0
          6300      CLRL14  EQU     $6300
          6540      MSG001  EQU     $6540

5F20                        ORG     $5F20


5F20 34   16        NORTHK  PSHS    A,B,X

5F22 BD   62C0              JSR     CLRL13  CLEAR LINE 13
5F25 BD   6300              JSR     CLRL14  CLEAR LINE 14

                    * CHECK FOR LEGAL MOVE
5F28 B6   5AE5              LDA     MXC     MX-COORDINATE
5F2B F6   5AE6              LDB     MYC     MY-COORDINATE
5F2E BD   5A80              JSR     MCSCCV  CONVERT TO
SX,SY
5F31 5A                     DECB            POINT TO NEXT
SY
5F32 BD   5B80              JSR     GTFVAL  GET THE FAKE
CHAR
5F35 C1   6C                CMPB    #HORDOR IS IT AN
OPENING
5F37 27   2C                BEQ     LBL002  GO IF YES
5F39 BD   6540              JSR     MSG001  DISPLAY ERROR
MESSAGE

                    * ADJUST STRENGTH VARIABLE
5F3C 34   06                PSHS    A,B     STRENGTH EFFECT
5F3E FC   549C              LDD     STRNTH
5F41 1083 0002              CMPD    #2      IS IT AT LIMIT
5F45 22   10                BHI     LBL001  GO IF NO
5F47 CC   0000              LDD     #0
5F4A FD   549C              STD     STRNTH
5F4D 35   06                PULS    A,B
5F4F BD   6400              JSR     RPTSTR  REPORT CURRENT
STRENGTH
```

```
5F52 35   16        00890          PULS    A,B,X
5F54 7E   64C0      00900          JMP     GMOVED  YOU DIED
                    00910
5F57 83   0002      00920 LBL001   SUBD    #2
5F5A FD   549C      00930          STD     STRNTH
5F5D 35   06        00940          PULS    A,B
5F5F BD   6400      00950          JSR     RPTSTR  REPORT CURRENT
STRENGTH
5F62 16   006E      00960          LBRA    LBL004  GO IF NOT AN
OPENING
                    00970
                    00980 * REVEAL THE CURRENT CELL
                    00990 * CONTENTS ON THE SCREEN.
                    01000 * (FROM UNDER THE AVATAR)
                    01010 * TESTING = USE SPACE
5F65 F6   5AE7      01020 LBL002   LDB     CELLCC  CONTENTS
5F68 4F             01030          CLRA            EXTEND IT
5F69 1F   01        01040          TFR     D,X
5F6B B6   5AE5      01050          LDA     MXC     MX-COORDINATE
5F6E F6   5AE6      01060          LDB     MYC     MY-COORDINATE
5F71 BD   5A80      01070          JSR     MCSCCV  CONVERT TO
SX,SY
5F74 BD   5300      01080          JSR     PTFCHR  PUT TO SCREEN
                    01090
                    01100 * GO ONE MAZE CELL NORTH
5F77 B6   5AE5      01110          LDA     MXC     CURRENT MX
5F7A B7   5AE8      01120          STA     MXN     NEW MX
5F7D F6   5AE6      01130          LDB     MYC     CURRENT MY
5F80 5A             01140          DECB
5F81 F7   5AE9      01150          STB     MYN     NEW MY
                    01160
                    01170 * SAVE NEW CELL SCREEN CONTENTS
5F84 BD   5A80      01180          JSR     MCSCCV  CONVERT TO
SX,SY
5F87 BD   5B80      01190          JSR     GTFVAL  GET CHAR VALUE
5F8A F7   5AE7      01200          STB     CELLCC  SAVE AS
CONTENTS
                    01210
                    01220 * PUT AVATAR TO NEW SCREEN LOCATION
5F8D B6   5AE8      01230          LDA     MXN     MX-COORDINATE
5F90 F6   5AE9      01240          LDB     MYN     MY-COORDINATE
5F93 BD   5A80      01250          JSR     MCSCCV  CONVERT TO
SX,SY
5F96 8E   0000      01260          LDX     #$0000  AVATAR CODE
EXTENDED
5F99 BD   5300      01270          JSR     PTFCHR  PUT TO SCREEN
                    01280
```

```
                 01290 * MAKE THE NEW COORDINATES CURRENT
5F9C B6   5AE8   01300        LDA     MXN     NEW MX
5F9F B7   5AE5   01310        STA     MXC     CURRENT MX
5FA2 F6   5AE9   01320        LDB     MYN     NEW MY
5FA5 F7   5AE6   01330        STB     MYC     CURRENT MY
                 01340
                 01350 * GO CHECK FOR PROVISIONS
5FA8 BD   5BA0   01360        JSR     PROCHK
                 01370
                 01380 * ADJUST STRENGTH VARIABLE
5FAB 34   06     01390        PSHS    A,B     STRENGTH EFFECT
5FAD FC   549C   01400        LDD     STRNTH
5FB0 1083 0001   01410        CMPD    #1      IS IT AT LIMIT
5FB4 22   12     01420        BHI     LBL003  GO IF NO
5FB6 CC   0000   01430        LDD     #0
5FB9 FD   549C   01440        STD     STRNTH
5FBC 35   06     01450        PULS    A,B
5FBE BD   6400   01460        JSR     RPTSTR  REPORT CURRENT
STRENGTH
5FC1 20   10     01470        BRA    LBL004
5FC3 35   16     01480        PULS    A,B,X
5FC5 7E   64C0   01490        JMP     GMOVED  YOU DIED
                 01500
5FC8 83   0001   01510 LBL003 SUBD    #1
5FCB FD   549C   01520        STD     STRNTH
5FCE 35   06     01530        PULS    A,B
5FD0 BD   6400   01540        JSR     RPTSTR  REPORT CURRENT
STRENGTH
                 01550
5FD3 35   16     01560 LBL004 PULS    A,B,X
5FD5 39          01570 ENDCHK RTS
                 01580
          0000   01590        END
```

=====

PCHARK: Pause Key (P-Key) Event Handler

The Assembly Language text listing:

```
                    00100 *****
                    00110 *
                    00120 * PCHARK.ASM
                    00130 * MDJ 2024/02/13
                    00140 *
                    00150 * PAUSE KEY
                    00160 * (P-KEY)
                    00170 * EVENT HANDLER
                    00180 *
                    00190 * NOT IMPLEMENTED FOR
                    00200 * MALKY'S WARREN
                    00210 *
                    00220 *****
                    00230
5FE0                00240         ORG     $5FE0
                    00250
5FE0 39             00260 PCHARK  RTS
                    00270
          0000      00280         END
```

=====

RCHARK: Resume Key (R-Key) Event Handler

The Assembly Language text listing:

```
                   00100 *****
                   00110 *
                   00120 * RCHARK.ASM
                   00130 * MDJ 2024/02/13
                   00140 *
                   00150 * RESUME KEY
                   00160 * (R-KEY)
                   00170 * EVENT HANDLER
                   00180 *
                   00190 * NOT IMPLEMENTED FOR
                   00200 * MALKY'S WARREN
                   00210 *
                   00220 *****
                   00230
6000               00240         ORG     $6000
                   00250
6000 39            00260 RCHARK  RTS
                   00270
          0000     00280         END
```

=====

SOUTHK: South Key (Down Arrow) Event Handler

The Assembly Language text listing:

```
         *****
         *
         * SOUTHK.ASM
         * MDJ 2024/02/13
         *
         * SOUTH KEY
         * (DOWN ARROW)
         * EVENT HANDLER
         *
         *****

         * PUT FAKE TEXT ROUTINE
5300     PTFCHR  EQU     $5300

         * COORDINATES CONVERTER
5A80     MCSCCV  EQU     $5A80

         * GET CHARACTER VALUE FROM
         * SCREEN INFORMATION BUFFERS
5B80     GTFVAL  EQU     $5B80

         * MAZE COORDINATES
         * AND CONTENTS
         * NOTE: THESE VARIABLES ARE
         *   INTERNAL TO STAVTR.ASM
5AE5     MXC     EQU     $5AE5
5AE6     MYC     EQU     $5AE6
5AE7     CELLCC  EQU     $5AE7
5AE8     MXN     EQU     $5AE8
5AE9     MYN     EQU     $5AE9

         * HORIZONTAL DOOR CODE
006C     HORDOR  EQU     $6C

         * GAME OVER ROUTINES
6500     GMOVER  EQU     $6500
64C0     GMOVED  EQU     $64C0

         * STRENGTH REPORTING
549C     STRNTH  EQU     $549C
```

```
          6400 RPTSTR  EQU     $6400

               * SCORE REPORTING
          549E SCORE   EQU     $549E
          6460 RPTSCO  EQU     $6460

          5BA0 PROCHK  EQU     $5BA0   PROVISIONS
CHECK

               * MESSAGE ROUTINES
          62C0 CLRL13  EQU     $62C0
          6300 CLRL14  EQU     $6300
          6540 MSG001  EQU     $6540

6020                   ORG     $6020

6020 34   16   SOUTHK  PSHS    A,B,X

6022 BD   62C0         JSR     CLRL13  CLEAR LINE 13
6025 BD   6300         JSR     CLRL14  CLEAR LINE 14

               * CHECK FOR LEGAL MOVE
6028 B6   5AE5         LDA     MXC     MX-COORDINATE
602B F6   5AE6         LDB     MYC     MY-COORDINATE
602E BD   5A80         JSR     MCSCCV  CONVERT TO
SX,SY
6031 5C                INCB            POINT TO NEXT
SY
6032 BD   5B80         JSR     GTFVAL  GET THE FAKE
CHAR
6035 C1   6C           CMPB    #HORDOR IS IT AN
OPENING
6037 27   2C           BEQ     LBL002  GO IF YES
6039 BD   6540         JSR     MSG001  DISPLAY ERROR
MESSAGE

               * ADJUST STRENGTH VARIABLE
603C 34   06           PSHS    A,B     STRENGTH EFFECT
603E FC   549C         LDD     STRNTH
6041 1083 0002         CMPD    #2      IS IT AT LIMIT
6045 22   10           BHI     LBL001  GO IF NO
6047 CC   0000         LDD     #0
604A FD   549C         STD     STRNTH
604D 35   06           PULS    A,B
604F BD   6400         JSR     RPTSTR  REPORT CURRENT
STRENGTH
6052 35   16           PULS    A,B,X
```

```
6054 7E   64C0     00900         JMP     GMOVED  YOU DIED
                   00910
6057 83   0002     00920 LBL001  SUBD    #2
605A FD   549C     00930         STD     STRNTH
605D 35   06       00940         PULS    A,B
605F BD   6400     00950         JSR     RPTSTR  REPORT CURRENT
STRENGTH
6062 16   006C     00960         LBRA    LBL004  GO IF NOT AN
OPENING
                   00970
                   00980 * REVEAL THE CURRENT CELL
                   00990 * CONTENTS ON THE SCREEN.
                   01000 * (FROM UNDER THE AVATAR)
                   01010 * TESTING = USE SPACE
6065 F6   5AE7     01020 LBL002  LDB     CELLCC  CONTENTS
6068 4F            01030         CLRA            EXTEND IT
6069 1F   01       01040         TFR     D,X
606B B6   5AE5     01050         LDA     MXC     MX-COORDINATE
606E F6   5AE6     01060         LDB     MYC     MY-COORDINATE
6071 BD   5A80     01070         JSR     MCSCCV  CONVERT TO
SX,SY
6074 BD   5300     01080         JSR     PTFCHR  PUT TO SCREEN
                   01090
                   01100 * GO ONE MAZE CELL SOUTH
6077 B6   5AE5     01110         LDA     MXC     CURRENT MX
607A B7   5AE8     01120         STA     MXN     NEW MX
607D F6   5AE6     01130         LDB     MYC     CURRENT MY
6080 5C            01140         INCB
6081 F7   5AE9     01150         STB     MYN     NEW MY
                   01160
                   01170 * SAVE NEW CELL SCREEN CONTENTS
6084 BD   5A80     01180         JSR     MCSCCV  CONVERT TO
SX,SY
6087 BD   5B80     01190         JSR     GTFVAL  GET CHAR VALUE
608A F7   5AE7     01200         STB     CELLCC  SAVE AS
CONTENTS
                   01210
                   01220 * PUT AVATAR TO NEW SCREEN LOCATION
608D B6   5AE8     01230         LDA     MXN     MX-COORDINATE
6090 F6   5AE9     01240         LDB     MYN     MY-COORDINATE
6093 BD   5A80     01250         JSR     MCSCCV  CONVERT TO
SX,SY
6096 8E   0000     01260         LDX     #$0000  AVATAR CODE
EXTENDED
6099 BD   5300     01270         JSR     PTFCHR  PUT TO SCREEN
                   01280
                   01290 * MAKE THE NEW COORDINATES CURRENT
```

```
609C B6   5AE8     01300          LDA     MXN     NEW MX
609F B7   5AE5     01310          STA     MXC     CURRENT MX
60A2 F6   5AE9     01320          LDB     MYN     NEW MY
60A5 F7   5AE6     01330          STB     MYC     CURRENT MY
                   01340
                   01350 * GO CHECK FOR PROVISIONS
60A8 BD   5BA0     01360          JSR     PROCHK
                   01370
                   01380 * ADJUST STRENGTH VARIABLE
60AB 34   06       01390          PSHS    A,B     STRENGTH EFFECT
60AD FC   549C     01400          LDD     STRNTH
60B0 1083 0001     01410          CMPD    #1      IS IT AT LIMIT
60B4 22   10       01420          BHI     LBL003  GO IF NO
60B6 CC   0000     01430          LDD     #0
60B9 FD   549C     01440          STD     STRNTH
60BC 35   06       01450          PULS    A,B
60BE BD   6400     01460          JSR     RPTSTR  REPORT CURRENT
STRENGTH
60C1 35   16       01470          PULS    A,B,X
60C3 7E   64C0     01480          JMP     GMOVED  YOU DIED
                   01490
60C6 83   0001     01500 LBL003   SUBD    #1
60C9 FD   549C     01510          STD     STRNTH
60CC 35   06       01520          PULS    A,B
60CE BD   6400     01530          JSR     RPTSTR  REPORT CURRENT
STRENGTH
                   01540
60D1 35   16       01550 LBL004   PULS    A,B,X
60D3 39            01560 ENDCHK   RTS
                   01570
          0000     01580          END
```

=====

TCHARK: Take Key (T-Key) Event Handler

The Assembly Language text listing:

```
                   *****
                   *
                   * TCHARK.ASM
                   * MDJ 2024/02/13
                   *
                   * TAKE KEY
                   * (T-KEY)
                   * (PUT CONTENTS OF CELL INTO BAG)
                   * EVENT HANDLER
                   *
                   *****

          54A1     BAG     EQU     $54A1   THE BAG
          64C0     GMOVED  EQU     $64C0   YOU DIED
          6300     CLRL14  EQU     $6300   CLEAR LINE 14
          6600     MSG004  EQU     $6600   "NOTHING HERE"
          6640     MSG005  EQU     $6640   "NO ROOM"
          66A0     MSG007  EQU     $66A0   "BAG CONTENTS"

                   * PUT FAKE TEXT ROUTINE
          5300     PTFCHR  EQU     $5300

                   * COORDINATES CONVERTER
          5A80     MCSCCV  EQU     $5A80

                   * PUT CHARACTER VALUE TO
                   * SCREEN INFORMATION BUFFERS
          5B60     PTFVAL  EQU     $5B60

                   * MAZE COORDINATES
                   * AND CONTENTS
          5AE5     MXC     EQU     $5AE5
          5AE6     MYC     EQU     $5AE6
          5AE7     CELLCC  EQU     $5AE7

                   * STRENGTH REPORTING
          549C     STRNTH  EQU     $549C
          6400     RPTSTR  EQU     $6400

60E0                       ORG     $60E0
```

```

60E0 34   16        TCHARK  PSHS    A,B,X

60E2 BD   6300              JSR     CLRL14  CLEAR LINE 14

                    * CHECK FOR GOSPEL OF JOHN
60E5 F6   5AE7              LDB     CELLCC  CURRENT CELL
CONTENTS
60E8 C1   E0                CMPB    #$E0    IS IT JOHN?
60EA 26   1C                BNE     LBLDC1  GO IF NO
60EC F7   54A1              STB     BAG     PUT IT IN THE
BAG
60EF C6   20                LDB     #$20    BLANK SPACE
60F1 F7   5AE7              STB     CELLCC  TO CURRENT
CONTENTS
60F4 4F                     CLRA            EXTEND IT
60F5 1F   01                TFR     D,X
60F7 B6   5AE5              LDA     MXC     MX-COORDINATE
60FA F6   5AE6              LDB     MYC     MY-COORDINATE
60FD BD   5A80              JSR     MCSCCV  CONVERT TO
SX,SY
6100 BD   5B60              JSR     PTFVAL  PUT TO SCREEN
BUFFER
6103 BD   66A0              JSR     MSG007  "BAG CONTENTS"
6106 20   2B                BRA     LBL002

6108 BD   6600      LBLDC1  JSR     MSG004  "NOTHING HERE"

                    * FAILED ACTION: ADJUST STRENGTH
VARIABLE
610B 34   06                PSHS    A,B     STRENGTH EFFECT
610D FC   549C              LDD     STRNTH
6110 1083 0002              CMPD    #2      IS IT AT LIMIT
6114 22   10                BHI     LBL001  GO IF NO
6116 CC   0000              LDD     #0
6119 FD   549C              STD     STRNTH
611C 35   06                PULS    A,B
611E BD   6400              JSR     RPTSTR  REPORT CURRENT
STRENGTH
6121 35   16                PULS    A,B,X
6123 7E   64C0              JMP     GMOVED  YOU DIED

6126 83   0002      LBL001  SUBD    #2
6129 FD   549C              STD     STRNTH
612C 35   06                PULS    A,B
612E BD   6400              JSR     RPTSTR  REPORT CURRENT
STRENGTH
```

```
6131 20   26        00890         BRA     LBLDC2
                    00900
                    00910 * COMPLETED ACTION: ADJUST STRENGTH
VARIABLE
6133 34   06        00920 LBL002  PSHS    A,B     STRENGTH EFFECT
6135 FC   549C      00930         LDD     STRNTH
6138 1083 0001      00940         CMPD    #1      IS IT AT LIMIT
613C 22   10        00950         BHI     LBL003  GO IF NO
613E CC   0000      00960         LDD     #0
6141 FD   549C      00970         STD     STRNTH
6144 35   06        00980         PULS    A,B
6146 BD   6400      00990         JSR     RPTSTR  REPORT CURRENT
STRENGTH
6149 35   16        01000         PULS    A,B,X
614B 7E   64C0      01010         JMP     GMOVED  YOU DIED
                    01020
614E 83   0001      01030 LBL003  SUBD    #1
6151 FD   549C      01040         STD     STRNTH
6154 35   06        01050         PULS    A,B
6156 BD   6400      01060         JSR     RPTSTR  REPORT CURRENT
STRENGTH
                    01070
6159 35   16        01080 LBLDC2  PULS    A,B,X
615B 39             01090 ENDCHK  RTS
                    01100
          0000      01110         END
```

=====

UCHARK: Up Key (U-Key) Event Handler

The Assembly Language text listing:

```
                   *****
                   *
                   * UCHARK.ASM
                   * MDJ 2024/02/13
                   *
                   * GO UP KEY
                   * (U-KEY)
                   * EVENT HANDLER
                   *
                   * NOT IMPLEMENTED FOR
                   * MALKY'S WARREN
                   *
                   *****

6160                       ORG     $6160

6160 39            UCHARK  RTS

          0000             END
```

=====

WESTK: West Key (Left Arrow) Event Handler

The Assembly Language text listing:

```
          *****
          *
          * WESTK.ASM
          * MDJ 2024/02/13
          *
          * WEST KEY
          * (LEFT ARROW)
          * EVENT HANDLER
          *
          *****

          * PUT FAKE TEXT ROUTINE
5300      PTFCHR  EQU     $5300

          * COORDINATES CONVERTER
5A80      MCSCCV  EQU     $5A80

          * GET CHARACTER VALUE FROM
          * SCREEN INFORMATION BUFFERS
5B80      GTFVAL  EQU     $5B80

          * MAZE COORDINATES
          * AND CONTENTS
          * NOTE: THESE VARIABLES ARE
          *   INTERNAL TO STAVTR.ASM
5AE5      MXC     EQU     $5AE5
5AE6      MYC     EQU     $5AE6
5AE7      CELLCC  EQU     $5AE7
5AE8      MXN     EQU     $5AE8
5AE9      MYN     EQU     $5AE9

          * VERTICAL DOOR CODE
0075      VRTDOR  EQU     $75

          * GAME OVER ROUTINES
6500      GMOVER  EQU     $6500
64C0      GMOVED  EQU     $64C0

          * STRENGTH REPORTING
549C      STRNTH  EQU     $549C
```

```
          6400     RPTSTR  EQU     $6400

                   * SCORE REPORTING
          549E     SCORE   EQU     $549E
          6460     RPTSCO  EQU     $6460

          5BA0     PROCHK  EQU     $5BA0   PROVISIONS
CHECK

                   * MESSAGE ROUTINES
          62C0     CLRL13  EQU     $62C0
          6300     CLRL14  EQU     $6300
          6540     MSG001  EQU     $6540

6180                       ORG     $6180

6180 34   16       WESTK   PSHS    A,B,X

6182 BD   62C0             JSR     CLRL13  CLEAR LINE 13
6185 BD   6300             JSR     CLRL14  CLEAR LINE 14

                   * CHECK FOR LEGAL MOVE
6188 B6   5AE5             LDA     MXC     MX-COORDINATE
618B F6   5AE6             LDB     MYC     MY-COORDINATE
618E BD   5A80             JSR     MCSCCV  CONVERT TO
SX,SY
6191 4A                    DECA            POINT TO NEXT
SX
6192 BD   5B80             JSR     GTFVAL  GET THE FAKE
CHAR
6195 C1   75               CMPB    #VRTDOR IS IT AN
OPENING
6197 27   2F               BEQ     LBL002  GO IF YES
6199 BD   6540             JSR     MSG001  DISPLAY ERROR
MESSAGE

                   * ADJUST STRENGTH VARIABLE
619C 34   06               PSHS    A,B     STRENGTH EFFECT
619E FC   549C             LDD     STRNTH
61A1 1083 0002             CMPD    #2      IS IT AT LIMIT
61A5 22   13               BHI     LBL001  GO IF NO
61A7 CC   0000             LDD     #0
61AA FD   549C             STD     STRNTH
61AD 35   06               PULS    A,B
61AF BD   6400             JSR     RPTSTR  REPORT CURRENT
STRENGTH
61B2 16   007F             LBRA    LBL004
```

```
61B5 35   16                 PULS    A,B,X
61B7 7E   64C0               JMP     GMOVED  YOU DIED

61BA 83   0002     LBL001    SUBD    #2
61BD FD   549C               STD     STRNTH
61C0 35   06                 PULS    A,B
61C2 BD   6400               JSR     RPTSTR  REPORT CURRENT
STRENGTH
61C5 16   006C               LBRA    LBL004  GO IF NOT AN
OPENING

                   * REVEAL THE CURRENT CELL
                   * CONTENTS ON THE SCREEN.
                   * (FROM UNDER THE AVATAR)
                   * TESTING = USE SPACE
61C8 F6   5AE7     LBL002    LDB     CELLCC  CONTENTS
61CB 4F                      CLRA            EXTEND IT
61CC 1F   01                 TFR     D,X
61CE B6   5AE5               LDA     MXC     MX-COORDINATE
61D1 F6   5AE6               LDB     MYC     MY-COORDINATE
61D4 BD   5A80               JSR     MCSCCV  CONVERT TO
SX,SY
61D7 BD   5300               JSR     PTFCHR  PUT TO SCREEN

                   * GO ONE MAZE CELL WEST
61DA B6   5AE5               LDA     MXC     CURRENT MX
61DD 4A                      DECA
61DE B7   5AE8               STA     MXN     NEW MX
61E1 F6   5AE6               LDB     MYC     CURRENT MY
61E4 F7   5AE9               STB     MYN     NEW MY

                   * SAVE NEW CELL SCREEN CONTENTS
61E7 BD   5A80               JSR     MCSCCV  CONVERT TO
SX,SY
61EA BD   5B80               JSR     GTFVAL  GET CHAR VALUE
61ED F7   5AE7               STB     CELLCC  SAVE AS
CONTENTS

                   * PUT AVATAR TO NEW SCREEN LOCATION
61F0 B6   5AE8               LDA     MXN     MX-COORDINATE
61F3 F6   5AE9               LDB     MYN     MY-COORDINATE
61F6 BD   5A80               JSR     MCSCCV  CONVERT TO
SX,SY
61F9 8E   0000               LDX     #$0000  AVATAR CODE
EXTENDED
61FC BD   5300               JSR     PTFCHR  PUT TO SCREEN

```

```
                   01300 * MAKE THE NEW COORDINATES CURRENT
61FF B6   5AE8     01310         LDA     MXN     NEW MX
6202 B7   5AE5     01320         STA     MXC     CURRENT MX
6205 F6   5AE9     01330         LDB     MYN     NEW MY
6208 F7   5AE6     01340         STB     MYC     CURRENT MY
                   01350
                   01360 * GO CHECK FOR PROVISIONS
620B BD   5BA0     01370         JSR     PROCHK
                   01380
                   01390 * ADJUST STRENGTH VARIABLE
620E 34   06       01400         PSHS    A,B     STRENGTH EFFECT
6210 FC   549C     01410         LDD     STRNTH
6213 1083 0001     01420         CMPD    #1      IS IT AT LIMIT
6217 22   10       01430         BHI     LBL003  GO IF NO
6219 CC   0000     01440         LDD     #0
621C FD   549C     01450         STD     STRNTH
621F 35   06       01460         PULS    A,B
6221 BD   6400     01470         JSR     RPTSTR  REPORT CURRENT
STRENGTH
6224 35   16       01480         PULS    A,B,X
6226 7E   64C0     01490         JMP     GMOVED  YOU DIED
                   01500
6229 83   0001     01510 LBL003  SUBD    #1
622C FD   549C     01520         STD     STRNTH
622F 35   06       01530         PULS    A,B
6231 BD   6400     01540         JSR     RPTSTR  REPORT CURRENT
STRENGTH
                   01550
6234 35   16       01560 LBL004  PULS    A,B,X
6236 39            01570 ENDCHK  RTS
                   01580
          0000     01590         END
```

=====

XCHARK: Exit Key (X-Key) Event Handler

The Assembly Language text listing:

```
                   *****
                   *
                   * XCHARK.ASM
                   * MDJ 2024/02/13
                   *
                   * EXIT KEY
                   * (X-KEY)
                   * EVENT HANDLER
                   *
                   * CHECKS FOR CONFIRMATION
                   * THEN EXITS GAME IF CONFIRMED
                   * (DOES A COLD START)
                   *
                   * RETURNS WITH NO ACTION
                   * IF NOT CONFIRMED
                   *
                   *****

                   * MLF POLCAT
          4142     POLCAT  EQU     $4142

                   * COLD START ADDRESS
          41A2     COLD    EQU     $41A2

                   * CONFIRM MESSAGE EQUATES
          62C0     CLRL13  EQU     $62C0   CLEAR LINE 13
          6300     CLRL14  EQU     $6300   CLEAR LINE 14
          6760     MSG010  EQU     $6760   "EXIT GAME
CONFIRM"

6240                       ORG     $6240

6240 34   02       XCHARK  PSHS    A

6242 BD   6300             JSR     CLRL14  GO CLEAR LINE
14
6245 BD   6760             JSR     MSG010  CONFIRM?

                   * GET A KEYPRESS
6248 BD   4142     LBL001  JSR     POLCAT
```

```
624B 27   FB        00480          BEQ     LBL001
                    00490
                    00500 * YCHARK (Y-KEY)
624D 81   59        00510          CMPA    #$59
624F 26   03        00520          BNE     LBL002
6251 7E   41A2      00530          JMP     COLD    GO DO COLD
START
                    00540
                    00550 * NCHARK (N-KEY)
6254 81   4E        00560 LBL002   CMPA    #$4E
6256 26   08        00570          BNE     LBL003
6258 BD   62C0      00580          JSR     CLRL13  GO CLEAR LINE
13
625B BD   6300      00590          JSR     CLRL14  GO CLEAR LINE
14
625E 20   03        00600          BRA     LBL004  GO DO RTS
                    00610
                    00620 * ANY OTHER KEYPRESS
6260 16   FFE5      00630 LBL003   LBRA    LBL001
                    00640
6263 35   02        00650 LBL004   PULS    A
6265 39             00660 ENDCHK   RTS
                    00670
          0000      00680          END
```

=====

YCHARK: “Yes” (Confirm) Key (Y-Key) Event Handler

The Assembly Language text listing:

```
                   00100 *****
                   00110 *
                   00120 * YCHARK.ASM
                   00130 * MDJ 2024/02/13
                   00140 *
                   00150 * CONFIRMED KEY
                   00160 * (Y-KEY)
                   00170 * EVENT HANDLER
                   00180 *
                   00190 * THIS IS AN UNUSED DUMMY
                   00200 * ROUTINE - FOR POTENTIAL
                   00210 * FUTURE USE ONLY
                   00220 *
                   00230 * WOULD DO A SIMPLE RETURN
                   00240 * TO CALLER TO VERIFY
                   00250 * THAT THE PROPOSED ACTION
                   00260 * IS INDEED CONFIRMED.
                   00270 *
                   00280 * NO ACTION IF Y-KEY IS
                   00290 * PRESSED IN GMLOOP.
                   00300 *
                   00310 *****
                   00320
6280               00330         ORG     $6280
                   00340
6280 39            00350 YCHARK  RTS
                   00360
          0000     00370         END
```

=====

CLRL13: Clear Line 13 of the Screen

The Assembly Language text listing:

```
                    00100 *****
                    00110 *
                    00120 * CLRL13.ASM
                    00130 * MDJ 2024/02/13
                    00140 *
                    00150 * CLEAR LINE 13
                    00160 *
                    00170 * ENTRY CONDITIONS
                    00180 * NONE
                    00190 *
                    00200 * EXIT CONDITIONS:
                    00210 * NONE
                    00220 *
                    00230 *****
                    00240
          63C0      00250 PTFSLS  EQU     $63C0
                    00260
62C0                00270         ORG     $62C0
                    00280
62C0 7E   62E5      00290 CLRL13  JMP     PTFL13
                    00300
62C3      0020      00310 LENL13  FDB    $0020   STRING LENGTH =
32
62C5      20        00320 TXTL13  FCC     "
"

          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
                   00330
62E5 34   36       00340 PTFL13  PSHS    A,B,X,Y
                   00350
62E7 86   00       00360         LDA     #0
62E9 C6   0D       00370         LDB     #13
62EB 8E   62C5     00380         LDX     #TXTL13
62EE 10BE 62C3     00390         LDY     LENL13
                   00400
62F2 BD   63C0     00410         JSR     PTFSLS
                   00420
62F5 35   36       00430         PULS    A,B,X,Y
62F7 39            00440 ENDCHK  RTS
                   00450
          0000     00460         END

     =====
```

CLRL14: Clear Line 14 of the Screen

The Assembly Language text listing:

```
                  00100 *****
                  00110 *
                  00120 * CLRL14.ASM
                  00130 * MDJ 2024/02/13
                  00140 *
                  00150 * CLEAR LINE 14
                  00160 *
                  00170 * ENTRY CONDITIONS
                  00180 * NONE
                  00190 *
                  00200 * EXIT CONDITIONS:
                  00210 * NONE
                  00220 *
                  00230 *****
                  00240
         63C0     00250 PTFSLS  EQU     $63C0
                  00260
6300              00270         ORG     $6300
                  00280
6300 7E  6325     00290 CLRL14  JMP     PTFL14
                  00300
6303     0020     00310 LENL14  FDB     $0020   STRING LENGTH =
32
6305     20       00320 TXTL14  FCC     "
"

         20
         20
         20
         20
         20
         20
         20
         20
         20
         20
         20
         20
         20
         20
         20
         20
```

```
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
          20
                   00330
6325 34   36       00340 PTFL14  PSHS    A,B,X,Y
                   00350
6327 86   00       00360         LDA     #0
6329 C6   0E       00370         LDB     #14
632B 8E   6305     00380         LDX     #TXTL14
632E 10BE 6303     00390         LDY     LENL14
                   00400
6332 BD   63C0     00410         JSR     PTFSLS
                   00420
6335 35   36       00430         PULS    A,B,X,Y
6337 39            00440 ENDCHK  RTS
                   00450
          0000     00460         END
```

=====

CLRSTR: Clear the Screen’s Strength Field

The Assembly Language text listing:

```
                  *****
                  *
                  * CLRSTR.ASM
                  * MDJ 2024/02/13
                  *
                  * BRUTE FORCE
                  * CLEAR THE
                  * STRENGTH FIELD
                  *
                  * ENTRY CONDITIONS
                  * NONE
                  *
                  * EXIT CONDITIONS:
                  * NONE
                  *
                  *****

          5300    PTFCHR  EQU     $5300

6340                      ORG     $6340

6340 34   16      CLRSTR  PSHS    A,B,X

6342 86   0B              LDA     #11
6344 C6   0F              LDB     #15
6346 8E   0020            LDX     #$0020
6349 BD   5300            JSR     PTFCHR
634C 86   0C              LDA     #12
634E C6   0F              LDB     #15
6350 8E   0020            LDX     #$0020
6353 BD   5300            JSR     PTFCHR
6356 86   0D              LDA     #13
6358 C6   0F              LDB     #15
635A 8E   0020            LDX     #$0020
635D BD   5300            JSR     PTFCHR
6360 86   0E              LDA     #14
6362 C6   0F              LDB     #15
6364 8E   0020            LDX     #$0020
6367 BD   5300            JSR     PTFCHR
636A 86   0F              LDA     #15
```

```
636C C6   0F       00500          LDB     #15
636E 8E   0020     00510          LDX     #$0020
6371 BD   5300     00520          JSR     PTFCHR
                   00530
6374 35   16       00540          PULS    A,B,X
6376 39            00550 ENDCHK   RTS
                   00560
          0000     00570          END
```

=====

CLRSCO: Clear the Screen’s Score Field

The Assembly Language text listing:

```
                    00100 *****
                    00110 *
                    00120 * CLRSCO.ASM
                    00130 * MDJ 2024/02/13
                    00140 *
                    00150 * BRUTE FORCE
                    00160 * CLEAR THE
                    00170 * SCORE FIELD
                    00180 *
                    00190 * ENTRY CONDITIONS
                    00200 * NONE
                    00210 *
                    00220 * EXIT CONDITIONS:
                    00230 * NONE
                    00240 *
                    00250 *****
                    00260
          5300      00270 PTFCHR  EQU     $5300
                    00280
6380                00290         ORG     $6380
                    00300
6380 34   16        00310 CLRSCO  PSHS    A,B,X
                    00320
6382 86   1B        00330         LDA     #27
6384 C6   0F        00340         LDB     #15
6386 8E   0020      00350         LDX     #$0020
6389 BD   5300      00360         JSR     PTFCHR
638C 86   1C        00370         LDA     #28
638E C6   0F        00380         LDB     #15
6390 8E   0020      00390         LDX     #$0020
6393 BD   5300      00400         JSR     PTFCHR
6396 86   1D        00410         LDA     #29
6398 C6   0F        00420         LDB     #15
639A 8E   0020      00430         LDX     #$0020
639D BD   5300      00440         JSR     PTFCHR
63A0 86   1E        00450         LDA     #30
63A2 C6   0F        00460         LDB     #15
63A4 8E   0020      00470         LDX     #$0020
63A7 BD   5300      00480         JSR     PTFCHR
63AA 86   1F        00490         LDA     #31
```

```
63AC C6   0F       00500         LDB     #15
63AE 8E   0020     00510         LDX     #$0020
63B1 BD   5300     00520         JSR     PTFCHR
                   00530
63B4 35   16       00540         PULS    A,B,X
63B6 39            00550 ENDCHK  RTS
                   00560
          0000     00570         END
```

=====

PTFSLS: Prints a Length-Specified FakeText String

The Assembly Language text listing:

```
                    *****
                    *
                    * PTFSLS.ASM
                    * MDJ 2024/02/16
                    *
                    * +++ FOR MAZE PROGRAMS ONLY +++
                    * +++    E.G.  MALKYS.BIN    +++
                    *
                    * PRINTS A LENGTH-SPECIFIED
                    * FAKETEXT STRING
                    * STARTING AT X (USUALLY = 0)
                    * AND Y (EITHER LINE 13 OR 14)
                    *
                    * ENTRY CONDITIONS:
                    * A = SCREEN X-COORDINATE (0-31)
                    * B = SCREEN Y-COORDINATE (13 OR 14 )
                    * X = START ADDRESS
                    *     OF THE STRING
                    * Y = STRING LENGTH
                    *     IN CHARACTERS
                    *     ($0001-$0020)
                    *     (    1-32   )
                    *
                    * +++   NO  ERROR CHECKING   +++
                    * THE USER IS RESPONSIBLE FOR ENSURING
                    * THAT THE ENTRY CONDITIONS ARE
CORRECT.
                    *
                    * EXIT CONDITIONS:
                    * NONE
                    *
                    *****

                    * EXTERNAL ROUTINE ADDRESS
          5300      PTFCHR  EQU     $5300

63C0                        ORG     $63C0

63C0 20   04        PTFSLS  BRA     LBL001

```

```
63C2                00490 TEMPA   RMB     1
63C3                00500 TEMPB   RMB     1
63C4                00510 TEMPX   RMB     2
                    00520
63C6 34   36        00530 LBL001  PSHS    A,B,X,Y
                    00540
63C8 B7   63C2      00550 LBL002  STA     TEMPA   SAVE SCREEN
INITIAL X-COORDINAT
E
63CB F7   63C3      00560         STB     TEMPB   SAVE SCREEN Y-
COORDINATE
63CE BF   63C4      00570         STX     TEMPX   SAVE THE
INITIAL CHARACTER POIN
TER
                    00580
63D1 BE   63C4      00590 LBL003  LDX     TEMPX   GET THE CURRENT
CHARACTER POINT
ER
                    00600
63D4 E6   80        00610         LDB     ,X+     GET A CHARACTER
FROM THE STRING
63D6 BF   63C4      00620         STX     TEMPX   SAVE THE NEXT
CHARACTER POINTER

63D9 4F             00630         CLRA            CLEAR D-
REGISTER HIGH BYTE
63DA 1F   01        00640         TFR     D,X     CHARACTER CODE
TO REGISTER X
                    00650
63DC B6   63C2      00660         LDA     TEMPA   GET SCREEN X-
COORDINATE
63DF F6   63C3      00670         LDB     TEMPB   GET SCREEN Y-
COORDINATE
63E2 BD   5300      00680         JSR     PTFCHR  PUT FAKE
CHARACTER TO SCREEN
                    00690
63E5 7C   63C2      00700         INC     TEMPA   INCREMENT
SCREEN X-COORDINATE
63E8 31   3F        00710         LEAY    -1,Y    DECREMENT
CHARACTER COUNTER
63EA 27   02        00720         BEQ     LBL004  GO IF ZERO (
==> DONE )
63EC 20   E3        00730         BRA     LBL003  RETURN FOR NEXT
CHARACTER
                    00740
63EE 35   36        00750 LBL004  PULS    A,B,X,Y
                    00760
```

```
63F0 39             00770 ENDCHK  RTS
                    00780
          0000      00790         END
```

=====

RPTSTR: Strength Reporter

The Assembly Language text listing:

```
                    *****
                    *
                    * RPTSTR.ASM
                    * MDJ 2024/02/13
                    *
                    * BRUTE FORCE
                    * STRENGTH REPORTER
                    *
                    * ENTRY CONDITIONS
                    * NONE
                    *
                    * EXIT CONDITIONS:
                    * NONE
                    *
                    *****

          5300      PTFCHR  EQU     $5300
          549C      STRNTH  EQU     $549C
          5A00      DECMAL  EQU     $5A00

                    * NOTE: THE FOLLOWING DIGITS
                    * ARE INTERNAL TO DECMAL.ASM
          5A04      DIGIT4  EQU     $5A04
          5A05      DIGIT3  EQU     $5A05
          5A06      DIGIT2  EQU     $5A06
          5A07      DIGIT1  EQU     $5A07
          5A08      DIGIT0  EQU     $5A08

6400                        ORG     $6400

6400 34   16        RPTSTR  PSHS    A,B,X

6402 FC   549C              LDD     STRNTH  GET CURRENT
STRENGTH
6405 BD   5A00              JSR     DECMAL  EXPRESS AS A
DECIMAL

6408 F6   5A04              LDB     DIGIT4  GET THE DIGIT
640B 4F                     CLRA            EXTEND IT
640C 1F   01                TFR     D,X     MOVE IT TO REG
X
640E 86   0B                LDA     #11     X-COORDINATE
```

```
6410 C6   0F       00490         LDB     #15      Y-COORDINATE
6412 BD   5300     00500         JSR     PTFCHR   REPORT IT
                   00510
6415 F6   5A05     00520         LDB     DIGIT3   GET THE DIGIT
6418 4F            00530         CLRA             EXTEND IT
6419 1F   01       00540         TFR     D,X      MOVE IT TO REG
X
641B 86   0C       00550         LDA     #12      X-COORDINATE
641D C6   0F       00560         LDB     #15      Y-COORDINATE
641F BD   5300     00570         JSR     PTFCHR   REPORT IT
                   00580
6422 F6   5A06     00590         LDB     DIGIT2   GET THE DIGIT
6425 4F            00600         CLRA             EXTEND IT
6426 1F   01       00610         TFR     D,X      MOVE IT TO REG
X
6428 86   0D       00620         LDA     #13      X-COORDINATE
642A C6   0F       00630         LDB     #15      Y-COORDINATE
642C BD   5300     00640         JSR     PTFCHR   REPORT IT
                   00650
642F F6   5A07     00660         LDB     DIGIT1   GET THE DIGIT
6432 4F            00670         CLRA             EXTEND IT
6433 1F   01       00680         TFR     D,X      MOVE IT TO REG
X
6435 86   0E       00690         LDA     #14      X-COORDINATE
6437 C6   0F       00700         LDB     #15      Y-COORDINATE
6439 BD   5300     00710         JSR     PTFCHR   REPORT IT
                   00720
643C F6   5A08     00730         LDB     DIGIT0   GET THE DIGIT
643F 4F            00740         CLRA             EXTEND IT
6440 1F   01       00750         TFR     D,X      MOVE IT TO REG
X
6442 86   0F       00760         LDA     #15      X-COORDINATE
6444 C6   0F       00770         LDB     #15      Y-COORDINATE
6446 BD   5300     00780         JSR     PTFCHR   REPORT IT
                   00790
6449 35   16       00800         PULS    A,B,X
644B 39            00810 ENDCHK  RTS
                   00820
          0000     00830         END
```

=====

RPTSCO: Score Reporter

The Assembly Language text listing:

```
                   *****
                   *
                   * RPTSCO.ASM
                   * MDJ 2024/02/13
                   *
                   * BRUTE FORCE
                   * SCORE REPORTER
                   *
                   * ENTRY CONDITIONS
                   * NONE
                   *
                   * EXIT CONDITIONS:
                   * NONE
                   *
                   *****

          5300     PTFCHR  EQU     $5300
          549E     SCORE   EQU     $549E
          5A00     DECMAL  EQU     $5A00

                   * NOTE: THE FOLLOWING DIGITS
                   * ARE INTERNAL TO DECMAL.ASM
          5A04     DIGIT4  EQU     $5A04
          5A05     DIGIT3  EQU     $5A05
          5A06     DIGIT2  EQU     $5A06
          5A07     DIGIT1  EQU     $5A07
          5A08     DIGIT0  EQU     $5A08

6460                       ORG     $6460

6460 34   16       RPTSTR  PSHS    A,B,X

6462 FC   549E             LDD     SCORE   GET CURRENT
STRENGTH
6465 BD   5A00             JSR     DECMAL  EXPRESS AS A
DECIMAL

6468 F6   5A04             LDB     DIGIT4  GET THE DIGIT
646B 4F                    CLRA            EXTEND IT
646C 1F   01               TFR     D,X     MOVE IT TO REG
X
646E 86   1B               LDA     #27     X-COORDINATE
```

```
6470 C6   0F       00490         LDB    #15     Y-COORDINATE
6472 BD   5300     00500         JSR    PTFCHR  REPORT IT
                   00510
6475 F6   5A05     00520         LDB    DIGIT3  GET THE DIGIT
6478 4F            00530         CLRA           EXTEND IT
6479 1F   01       00540         TFR    D,X     MOVE IT TO REG
X
647B 86   1C       00550         LDA    #28     X-COORDINATE
647D C6   0F       00560         LDB    #15     Y-COORDINATE
647F BD   5300     00570         JSR    PTFCHR  REPORT IT
                   00580
6482 F6   5A06     00590         LDB    DIGIT2  GET THE DIGIT
6485 4F            00600         CLRA           EXTEND IT
6486 1F   01       00610         TFR    D,X     MOVE IT TO REG
X
6488 86   1D       00620         LDA    #29     X-COORDINATE
648A C6   0F       00630         LDB    #15     Y-COORDINATE
648C BD   5300     00640         JSR    PTFCHR  REPORT IT
                   00650
648F F6   5A07     00660         LDB    DIGIT1  GET THE DIGIT
6492 4F            00670         CLRA           EXTEND IT
6493 1F   01       00680         TFR    D,X     MOVE IT TO REG
X
6495 86   1E       00690         LDA    #30     X-COORDINATE
6497 C6   0F       00700         LDB    #15     Y-COORDINATE
6499 BD   5300     00710         JSR    PTFCHR  REPORT IT
                   00720
649C F6   5A08     00730         LDB    DIGIT0  GET THE DIGIT
649F 4F            00740         CLRA           EXTEND IT
64A0 1F   01       00750         TFR    D,X     MOVE IT TO REG
X
64A2 86   1F       00760         LDA    #31     X-COORDINATE
64A4 C6   0F       00770         LDB    #15     Y-COORDINATE
64A6 BD   5300     00780         JSR    PTFCHR  REPORT IT
                   00790
64A9 35   16       00800         PULS   A,B,X
64AB 39            00810 ENDCHK  RTS
                   00820
          0000     00830         END
```

=====

GMOVED: Game Over - You Died

The Assembly Language text listing:

```
                    00100 *****
                    00110 *
                    00120 * GMOVED.ASM
                    00130 * MDJ 2024/02/13
                    00140 *
                    00150 * GAME OVER
                    00160 * YOU DIED
                    00170 *
                    00180 *****
                    00190
          54A0      00200 GMOK    EQU     $54A0   GAME STATUS
FLAG
          6580      00210 MSG002  EQU     $6580   YOU DIED MSG
          4142      00220 POLCAT  EQU     $4142
          6240      00230 XCHARK  EQU     $6240
          5DE0      00240 GCHARK  EQU     $5DE0
                    00250
64C0                00260         ORG     $64C0
                    00270
64C0 34   02        00280 GMOVED  PSHS    A
                    00290
64C2 86   00        00300         LDA     #0      GAME OVER CODE
64C4 B7   54A0      00310         STA     GMOK    PUT TO STATUS
FLAG
64C7 BD   6580      00320         JSR     MSG002  DISPLAY MESSAGE
                    00330
                    00340 * GET A KEYPRESS
64CA BD   4142      00350 LBL001  JSR     POLCAT
64CD 27   FB        00360         BEQ     LBL001
                    00370
                    00380 * XCHARK (X-KEY = EXIT GAME)
64CF 81   58        00390 LBL002  CMPA    #$58
64D1 26   05        00400         BNE     LBL003
64D3 BD   6240      00410         JSR     XCHARK  JUMPS TO COLD
64D6 20   F2        00420         BRA     LBL001  DUMMY
                    00430
                    00440 * GCHARK (G-KEY = NEW GAME)
64D8 81   47        00450 LBL003  CMPA    #$47
64DA 26   05        00460         BNE     LBL004
64DC BD   5DE0      00470         JSR     GCHARK  JUMPS TO SMREAD
64DF 20   E9        00480         BRA     LBL001  DUMMY
                    00490
                    00500 * ANY OTHER KEYPRESS
64E1 20   E7        00510 LBL004   BRA    LBL001
                    00520
64E3 35   02        00530 GDEXIT  PULS    A
64E5 39             00540 ENDCHK  RTS
                    00550
          0000      00560         END
```

=====

GMOVER: Game Over - Quest Complete

The Assembly Language text listing:

```
                    *****
                    *
                    * GMOVER.ASM
                    * MDJ 2024/02/13
                    *
                    * GAME OVER
                    * QUEST COMPLETE
                    *
                    *****

          54A0      GMOK    EQU     $54A0   GAME STATUS
FLAG
          65C0      MSG003  EQU     $65C0   QUEST COMPLETE
MSG
          4142      POLCAT  EQU     $4142
          6240      XCHARK  EQU     $6240
          5DE0      GCHARK  EQU     $5DE0

6500                        ORG     $6500

6500 34   02        GMOVER  PSHS    A

6502 86   00                LDA     #0      GAME OVER CODE
6504 B7   54A0              STA     GMOK    PUT TO STATUS
FLAG
6507 BD   65C0              JSR     MSG003  DISPLAY MESSAGE

                    * GET A KEYPRESS
650A BD   4142      LBL001  JSR     POLCAT
650D 27   FB                BEQ     LBL001

                    * XCHARK (X-KEY = EXIT GAME)
650F 81   58        LBL002  CMPA    #$58
6511 26   05                BNE     LBL003
6513 BD   6240              JSR     XCHARK  JUMPS TO COLD
6516 20   F2                BRA     LBL001  DUMMY

                    * GCHARK (G-KEY = NEW GAME)
6518 81   47        LBL003  CMPA    #$47
651A 26   05                BNE     LBL004
651C BD   5DE0              JSR     GCHARK  JUMPS TO SMREAD
651F 20   E9                BRA     LBL001  DUMMY
```

```
                    00490
                    00500 * ANY OTHER KEYPRESS
6521 20   E7        00510 LBL004   BRA    LBL001
                    00520
6523 35   02        00530 GREXIT  PULS    A
6525 39             00540 ENDCHK  RTS
                    00550
          0000      00560         END
```

=====

MSG001: “You Can’t Go That Way”

The Assembly Language text listing:

```
                   00100 *****
                   00110 *
                   00120 * MSG001.ASM
                   00130 * MDJ 2024/02/13
                   00140 *
                   00150 * MESSAGE NO. 001
                   00160 *
                   00170 * ENTRY CONDITIONS
                   00180 * NONE
                   00190 *
                   00200 * EXIT CONDITIONS:
                   00210 * NONE
                   00220 *
                   00230 *****
                   00240
          63C0     00250 PTFSLS  EQU     $63C0
                   00260
6540               00270         ORG     $6540
                   00280
6540 7E   655A     00290 MSG001  JMP     PTF001
                   00300
6543      0015     00310 LEN001  FDB     $0015   STRING LENGTH =
21
6545      59       00320 TXT001  FCC     "YOU CAN'T GO THAT WAY"
          4F
          55
          20
          43
          41
          4E
          27
          54
          20
          47
          4F
          20
          54
          48
          41
          54
          20
          57
          41
          59
                   00330
655A 34   36       00340 PTF001  PSHS    A,B,X,Y
                   00350
655C 86   00       00360         LDA     #0
655E C6   0E       00370         LDB     #14
6560 8E   6545     00380         LDX     #TXT001
6563 10BE 6543     00390         LDY     LEN001
                   00400
6567 BD   63C0     00410         JSR     PTFSLS
                   00420
656A 35   36       00430         PULS    A,B,X,Y
656C 39            00440 ENDCHK  RTS
                   00450
          0000     00460         END

     =====
```

MSG002: “ ** Game Over: You Died!”

The Assembly Language text listing:

```
                    *****
                    *
                    * MSG002.ASM
                    * MDJ 2024/02/13
                    *
                    * MESSAGE NO. 002
                    *
                    * ENTRY CONDITIONS
                    * NONE
                    *
                    * EXIT CONDITIONS:
                    * NONE
                    *
                    *****

          63C0      PTFSLS  EQU     $63C0

6580                        ORG     $6580

6580 7E   659E      MSG002  JMP     PTF002

6583      0019      LEN002  FDB     $0019   STRING LENGTH =
25
6585      20        TXT002  FCC     "  ** GAME OVER: YOU
DIED!"
          20
          2A
          2A
          20
          47
          41
          4D
          45
          20
          4F
          56
          45
          52
          3A
          20
          59
          4F
          55
          20
          44
          49
          45
          44
          21
                   00330
659E 34   36       00340 PTF002  PSHS    A,B,X,Y
                   00350
65A0 86   00       00360         LDA     #0
65A2 C6   0D       00370         LDB     #13
65A4 8E   6585     00380         LDX     #TXT002
65A7 10BE 6583     00390         LDY     LEN002
                   00400
65AB BD   63C0     00410         JSR     PTFSLS
                   00420
65AE 35   36       00430         PULS    A,B,X,Y
65B0 39            00440 ENDCHK  RTS
                   00450
          0000     00460         END
```

=====

MSG003:
“ ** Game Over: Quest Complete.”

The Assembly Language text listing:

```
                   *****
                   *
                   * MSG003.ASM
                   * MDJ 2024/02/13
                   *
                   * MESSAGE NO. 003
                   *
                   * ENTRY CONDITIONS
                   * NONE
                   *
                   * EXIT CONDITIONS:
                   * NONE
                   *
                   *****

          63C0     PTFSLS  EQU     $63C0

65C0                       ORG     $65C0

65C0 7E   65E4     MSG003  JMP     PTF003

65C3      001F     LEN003  FDB     $001F   STRING LENGTH =
31
65C5      20       TXT003  FCC     "  ** GAME OVER: QUEST
COMPLETE."

          20
          2A
          2A
          20
          47
          41
          4D
          45
          20
          4F
          56
          45
          52
          3A
```

```
           20
           51
           55
           45
           53
           54
           20
           43
           4F
           4D
           50
           4C
           45
           54
           45
           2E
                     00330
65E4 34    36        00340 PTF003  PSHS    A,B,X,Y
                     00350
65E6 86    00        00360         LDA     #0
65E8 C6    0D        00370         LDB     #13
65EA 8E    65C5      00380         LDX     #TXT003
65ED 10BE  65C3      00390         LDY     LEN003
                     00400
65F1 BD    63C0      00410         JSR     PTFSLS
                     00420
65F4 35    36        00430         PULS    A,B,X,Y
65F6 39              00440 ENDCHK  RTS
                     00450
           0000      00460         END
```

=====

MSG004: “There’s Nothing Here.”

The Assembly Language text listing:

```
                    00100 *****
                    00110 *
                    00120 * MSG004.ASM
                    00130 * MDJ 2024/02/13
                    00140 *
                    00150 * MESSAGE NO. 004
                    00160 *
                    00170 * ENTRY CONDITIONS
                    00180 * NONE
                    00190 *
                    00200 * EXIT CONDITIONS:
                    00210 * NONE
                    00220 *
                    00230 *****
                    00240
          63C0      00250 PTFSLS  EQU     $63C0
                    00260
6600                00270         ORG     $6600
                    00280
6600 7E   661A      00290 MSG004  JMP     PTF004
                    00300
6603      0015      00310 LEN004  FDB     $0015   STRING LENGTH =
21
6605      54        00320 TXT004  FCC     "THERE'S NOTHING HERE."
          48
          45
          52
          45
          27
          53
          20
          4E
          4F
          54
          48
          49
          4E
          47
          20
          48
          45
          52
          45
          2E
                   00330
661A 34   36       00340 PTF004  PSHS    A,B,X,Y
                   00350
661C 86   00       00360         LDA     #0
661E C6   0E       00370         LDB     #14
6620 8E   6605     00380         LDX     #TXT004
6623 10BE 6603     00390         LDY     LEN004
                   00400
6627 BD   63C0     00410         JSR     PTFSLS
                   00420
662A 35   36       00430         PULS    A,B,X,Y
662C 39            00440 ENDCHK  RTS
                   00450
          0000     00460         END

     =====
```

MSG005: “No Room.”

The Assembly Language text listing:

```
                   00100 *****
                   00110 *
                   00120 * MSG005.ASM
                   00130 * MDJ 2024/02/13
                   00140 *
                   00150 * MESSAGE NO. 005
                   00160 *
                   00170 * ENTRY CONDITIONS
                   00180 * NONE
                   00190 *
                   00200 * EXIT CONDITIONS:
                   00210 * NONE
                   00220 *
                   00230 *****
                   00240
          63C0     00250 PTFSLS  EQU     $63C0
                   00260
6640               00270         ORG     $6640
                   00280
6640 7E   664D     00290 MSG005  JMP     PTF005
                   00300
6643      0008     00310 LEN005  FDB     $0008   STRING LENGTH =
8
6645      4E       00320 TXT005  FCC     "NO ROOM."
          4F
          20
          52
          4F
          4F
          4D
          2E
                   00330
664D 34   36       00340 PTF005  PSHS    A,B,X,Y
                   00350
664F 86   00       00360         LDA     #0
6651 C6   0E       00370         LDB     #14
6653 8E   6645     00380         LDX     #TXT005
6656 10BE 6643     00390         LDY     LEN005
                   00400
665A BD   63C0     00410         JSR     PTFSLS
                   00420
665D 35   36       00430         PULS    A,B,X,Y
```

```
665F 39             00440 ENDCHK  RTS
                    00450
          0000      00460         END
```

=====

MSG006: “The Bag is Empty.”

The Assembly Language text listing:

```
                   *****
                   *
                   * MSG006.ASM
                   * MDJ 2024/02/13
                   *
                   * MESSAGE NO. 006
                   *
                   * ENTRY CONDITIONS
                   * NONE
                   *
                   * EXIT CONDITIONS:
                   * NONE
                   *
                   *****

          63C0     PTFSLS  EQU     $63C0

6660                       ORG     $6660

6660 7E   6676     MSG006  JMP     PTF006

6663      0011     LEN006  FDB     $0011   STRING LENGTH =
17
6665      54       TXT006  FCC     "THE BAG IS EMPTY."
          48
          45
          20
          42
          41
          47
          20
          49
          53
          20
          45
          4D
          50
          54
          59
          2E

6676 34   36       PTF006  PSHS    A,B,X,Y
```

```
                    00350
6678 86   00        00360          LDA     #0
667A C6   0E        00370          LDB     #14
667C 8E   6665      00380          LDX     #TXT006
667F 10BE 6663      00390          LDY     LEN006
                    00400
6683 BD   63C0      00410          JSR     PTFSLS
                    00420
6686 35   36        00430          PULS    A,B,X,Y
6688 39             00440 ENDCHK   RTS
                    00450
          0000      00460          END
```

=====

MSG007:
“Bag Contents: Gospel of John.”

The Assembly Language text listing:

```
                  00100 *****
                  00110 *
                  00120 * MSG007.ASM
                  00130 * MDJ 2024/02/13
                  00140 *
                  00150 * MESSAGE NO. 007
                  00160 *
                  00170 * ENTRY CONDITIONS
                  00180 * NONE
                  00190 *
                  00200 * EXIT CONDITIONS:
                  00210 * NONE
                  00220 *
                  00230 *****
                  00240
         63C0     00250 PTFSLS  EQU     $63C0
                  00260
66A0              00270         ORG     $66A0
                  00280
66A0 7E  66C2     00290 MSG007  JMP     PTF007
                  00300
66A3     001D     00310 LEN007  FDB     $001D   STRING LENGTH =
29
66A5     42       00320 TXT007  FCC     "BAG CONTENTS: GOSPEL
OF JOHN."

         41
         47
         20
         43
         4F
         4E
         54
         45
         4E
         54
         53
         3A
         20
         47
```

```
          4F
          53
          50
          45
          4C
          20
          4F
          46
          20
          4A
          4F
          48
          4E
          2E
                   00330
66C2 34   36       00340 PTF007  PSHS    A,B,X,Y
                   00350
66C4 86   00       00360         LDA     #0
66C6 C6   0E       00370         LDB     #14
66C8 8E   66A5     00380         LDX     #TXT007
66CB 10BE 66A3     00390         LDY     LEN007
                   00400
66CF BD   63C0     00410         JSR     PTFSLS
                   00420
66D2 35   36       00430         PULS    A,B,X,Y
66D4 39            00440 ENDCHK  RTS
                   00450
          0000     00460         END
```

=====

MSG008: “The Warehouse is Empty.”

The Assembly Language text listing:

```
                  00100 *****
                  00110 *
                  00120 * MSG008.ASM
                  00130 * MDJ 2024/02/13
                  00140 *
                  00150 * MESSAGE NO. 008
                  00160 *
                  00170 * ENTRY CONDITIONS
                  00180 * NONE
                  00190 *
                  00200 * EXIT CONDITIONS:
                  00210 * NONE
                  00220 *
                  00230 *****
                  00240
          63C0    00250 PTFSLS  EQU     $63C0
                  00260
66E0              00270         ORG     $66E0
                  00280
66E0 7E   66FC    00290 MSG008  JMP     PTF008
                  00300
66E3      0017    00310 LEN008  FDB     $0017   STRING LENGTH =
23
66E5      54      00320 TXT008  FCC     "THE WAREHOUSE IS
EMPTY."
          48
          45
          20
          57
          41
          52
          45
          48
          4F
          55
          53
          45
          20
          49
          53
          20
          45
          4D
          50
          54
          59
          2E
                   00330
66FC 34   36       00340 PTF008  PSHS    A,B,X,Y
                   00350
66FE 86   00       00360         LDA     #0
6700 C6   0E       00370         LDB     #14
6702 8E   66E5     00380         LDX     #TXT008
6705 10BE 66E3     00390         LDY     LEN008
                   00400
6709 BD   63C0     00410         JSR     PTFSLS
                   00420
670C 35   36       00430         PULS    A,B,X,Y
670E 39            00440 ENDCHK  RTS
                   00450
          0000     00460         END
```

=====

MSG009:
“Whse Inventory: Gospel of John.”

The Assembly Language text listing:

```
                   *****
                   *
                   * MSG009.ASM
                   * MDJ 2024/02/13
                   *
                   * MESSAGE NO. 009
                   *
                   * ENTRY CONDITIONS
                   * NONE
                   *
                   * EXIT CONDITIONS:
                   * NONE
                   *
                   *****

          63C0     PTFSLS  EQU     $63C0

6720                       ORG     $6720

6720 7E   6744     MSG009  JMP     PTF009

6723      001F     LEN009  FDB     $001F   STRING LENGTH =
31
6725      57       TXT009  FCC     "WHSE INVENTORY: GOSPEL
OF JOHN."

          48
          53
          45
          20
          49
          4E
          56
          45
          4E
          54
          4F
          52
          59
          3A
          20
          47
          4F
          53
          50
          45
          4C
          20
          4F
          46
          20
          4A
          4F
          48
          4E
          2E
                   00330
6744 34   36       00340 PTF009  PSHS    A,B,X,Y
                   00350
6746 86   00       00360         LDA     #0
6748 C6   0E       00370         LDB     #14
674A 8E   6725     00380         LDX     #TXT009
674D 10BE 6723     00390         LDY     LEN009
                   00400
6751 BD   63C0     00410         JSR     PTFSLS
                   00420
6754 35   36       00430         PULS    A,B,X,Y
6756 39            00440 ENDCHK  RTS
                   00450
          0000     00460         END
```

=====

MSG010:
“Exit Game Confirm? - Y or N”

The Assembly Language text listing:

```
                   00100 *****
                   00110 *
                   00120 * MSG010.ASM
                   00130 * MDJ 2024/02/13
                   00140 *
                   00150 * MESSAGE NO. 010
                   00160 *
                   00170 * ENTRY CONDITIONS
                   00180 * NONE
                   00190 *
                   00200 * EXIT CONDITIONS:
                   00210 * NONE
                   00220 *
                   00230 *****
                   00240
          63C0     00250 PTFSLS  EQU     $63C0
                   00260
6760               00270         ORG     $6760
                   00280
6760 7E   6780     00290 MSG010  JMP     PTF010
                   00300
6763      001B     00310 LEN010  FDB     $001B   STRING LENGTH =
27
6765      45       00320 TXT010  FCC     "EXIT GAME CONFIRM? - Y
OR N"
          58
          49
          54
          20
          47
          41
          4D
          45
          20
          43
          4F
          4E
          46
          49
          52
          4D
          3F
          20
          2D
          20
          59
          20
          4F
          52
          20
          4E
                   00330
6780 34   36       00340 PTF010  PSHS    A,B,X,Y
                   00350
6782 86   00       00360         LDA     #0
6784 C6   0E       00370         LDB     #14
6786 8E   6765     00380         LDX     #TXT010
6789 10BE 6763     00390         LDY     LEN010
                   00400
678D BD   63C0     00410         JSR     PTFSLS
                   00420
6790 35   36       00430         PULS    A,B,X,Y
6792 39            00440 ENDCHK  RTS
                   00450
          0000     00460         END
```

=====

MSG011:
"New Game Confirm? - Y or N"

The Assembly Language text listing:

```
                    00100 *****
                    00110 *
                    00120 * MSG011.ASM
                    00130 * MDJ 2024/02/13
                    00140 *
                    00150 * MESSAGE NO. 011
                    00160 *
                    00170 * ENTRY CONDITIONS
                    00180 * NONE
                    00190 *
                    00200 * EXIT CONDITIONS:
                    00210 * NONE
                    00220 *
                    00230 *****
                    00240
          63C0      00250 PTFSLS  EQU     $63C0
                    00260
67A0                00270         ORG     $67A0
                    00280
67A0 7E   67BF      00290 MSG011  JMP     PTF011
                    00300
67A3      001A      00310 LEN011  FDB     $001A   STRING LENGTH =
26
67A5      4E        00320 TXT011  FCC     "NEW GAME CONFIRM? - Y
OR N"
          45
          57
          20
          47
          41
          4D
          45
          20
          43
          4F
          4E
          46
          49
          52
          4D
```

```
           3F
           20
           2D
           20
           59
           20
           4F
           52
           20
           4E
                    00330
67BF 34    36       00340 PTF011  PSHS    A,B,X,Y
                    00350
67C1 86    00       00360         LDA     #0
67C3 C6    0E       00370         LDB     #14
67C5 8E    67A5     00380         LDX     #TXT011
67C8 10BE  67A3     00390         LDY     LEN011
                    00400
67CC BD    63C0     00410         JSR     PTFSLS
                    00420
67CF 35    36       00430         PULS    A,B,X,Y
67D1 39             00440 ENDCHK  RTS
                    00450
           0000     00460         END
```

=====

SMGAME: Displays the Maze and Starts the Game

The Assembly Language text listing:

```
                  00100 *****
                  00110 *
                  00120 * SMGAME.ASM
                  00130 * MDJ 2024/02/13
                  00140 *
                  00150 * SCREEN MAZE GAME
                  00160 * ASSEMBLY ROUTINE
                  00170 *
                  00180 * DISPLAYS THE MAZE
                  00190 * ON SCREEN, USING THE
                  00200 * FAKETEXT 32 X 16
                  00210 * CHARACTER SET FOR
                  00220 * PMODE 4, AND THEN
                  00230 * STARTS THE GAME.
                  00240 *
                  00250 *****
                  00260
          4142    00270 POLCAT  EQU    $4142   GET A KEYPRESS
          43D7    00280 RSEED   EQU    $43D7   SET RANDOM SEED
          5300    00290 PTFCHR  EQU    $5300   FAKE TEXT
ROUTINE
          5500    00300 LINE00  EQU    $5500   START OF
BUFFERS
          5A80    00310 MCSCCV  EQU    $5A80   COORDINATE
CONVERTER
          5AE0    00320 STAVTR  EQU    $5AE0   AVATAR INITIAL
SETUP
          5B80    00330 GTFVAL  EQU    $5B80   GET CELL
CONTENTS
          5B60    00340 PTFVAL  EQU    $5B60   PUT CELL
CONTENTS
          549C    00350 STRNTH  EQU    $549C   STRENGTH SYSTEM
VARIABLE
          549E    00360 SCORE   EQU    $549E   SCORE SYSTEM
VARIABLE
          54A0    00370 GMOK    EQU    $54A0   GAME STATUS
FLAG
          54A1    00380 BAG     EQU    $54A1   BAG CONTENTS
          54A2    00390 WHSE    EQU    $54A2   WAREHOUSE
CONTENTS
```

```
          54A3      00400 DOCVAL  EQU     $54A3   DOCUMENT VALUE
          54A4      00410 PROVAL  EQU     $54A4   PROVISIONS
VALUE
          6960      00420 GMLOOP  EQU     $6960   GAME LOOP
          6400      00430 RPTSTR  EQU     $6400   REPORT STRENGTH
          6460      00440 RPTSCO  EQU     $6460   REPORT SCORE
          5A60      00450 RANDOM  EQU     $5A60   RANDOM NUMBER
GENERATOR
                    00460
67E0                00470         ORG     $67E0
                    00480
67E0 7E   67E8      00490 SMGAME  JMP     LBL001
                    00500
67E3                00510 XCOORD  RMB     1
67E4                00520 YCOORD  RMB     1
67E5                00530 CHRCOD  RMB     2
67E7                00540 RANGE   RMB     1
                    00550
67E8 34   76        00560 LBL001  PSHS    A,B,X,Y,U
                    00570
                    00580 * INITIALIZE SYSTEM VARIABLES
67EA BD   43D7      00590         JSR     RSEED   SET RANDOM SEED
67ED 8E   0064      00600         LDX     #100
67F0 BF   549C      00610         STX     STRNTH  INITIAL
STRENGTH
67F3 8E   0000      00620         LDX     #0
67F6 BF   549E      00630         STX     SCORE   INITIAL SCORE
67F9 86   01        00640         LDA     #1      GAME IS RUNNING
67FB B7   54A0      00650         STA     GMOK    GAME STATUS
FLAG
67FE 86   20        00660         LDA     #$20    "EMPTY" CODE
6800 B7   54A1      00670         STA     BAG     BAG CONTENTS
6803 B7   54A2      00680         STA     WHSE    WAREHOUSE
CONTENTS
6806 CC   0005      00690         LDD     #5      MAXIMUM FOR
RANDOM
6809 BD   5A60      00700         JSR     RANDOM  RANDOM #
GENERATOR
680C 86   15        00710         LDA     #21     # OF JOHN
CHAPTERS
680E 3D             00720         MUL
680F C3   0069      00730         ADDD    #105    MINIMUM DOCVAL
6812 F7   54A3      00740         STB     DOCVAL  DOCUMENT VALUE
6815 CC   0032      00750         LDD     #50     MAXIMUM FOR
RANDOM
6818 BD   5A60      00760         JSR     RANDOM  RANDOM #
GENERATOR
```

```
681B C3   0019     00770          ADDD    #25     MINIMUM PROVAL
681E F7   54A4     00780          STB     PROVAL  PROVISIONS
VALUE
                   00790
                   00800 * INITIALIZE THE DOCUMENT LOCATION
6821 8E   00E0     00810          LDX     #$00E0  CHRCOD FOR JOHN
6824 BF   67E5     00820          STX     CHRCOD
6827 CC   0005     00830          LDD     #5      MAXIMUM MY
682A BD   5A60     00840          JSR     RANDOM  RANDOM #
GENERATOR
682D F7   67E4     00850          STB     YCOORD  MY-COORDINATE
6830 C1   00       00860          CMPB    #0      IS IT FIRST
ROW?
6832 27   0C       00870          BEQ     LBL005  GO IF YES
6834 C1   05       00880          CMPB    #5      IS IT LAST ROW?
6836 27   13       00890          BEQ     LBL006  GO IF YES
6838 CC   000E     00900          LDD     #14     RANGE FOR ROWS
1-4
683B BD   5A60     00910          JSR     RANDOM  RANDOM #
GENERATOR
683E 20   11       00920          BRA     LBL007
6840 CC   0009     00930 LBL005   LDD     #9      RANGE FOR ROW 0
6843 BD   5A60     00940          JSR     RANDOM  RANDOM #
GENERATOR
6846 C3   0005     00950          ADDD    #5      OFFSET FOR ROW
0
6849 20   06       00960          BRA     LBL007
684B CC   0009     00970 LBL006   LDD     #9      RANGE FOR ROW 5
684E BD   5A60     00980          JSR     RANDOM  RANDOM #
GENERATOR
6851 F7   67E3     00990 LBL007   STB     XCOORD  MX-COORDINATE
6854 B6   67E3     01000          LDA     XCOORD  MX-COORDINATE
6857 F6   67E4     01010          LDB     YCOORD  MY-COORDINATE
685A BD   5A80     01020          JSR     MCSCCV  CONVERT TO
(SX,SY)
685D BE   67E5     01030          LDX     CHRCOD  CHARACTER CODE
6860 BD   5B60     01040          JSR     PTFVAL  PUT TO SCREEN
BUFFER
                   01050
                   01060 * INITIALIZE THE PROVISIONS LOCATION
6863 8E   0085     01070          LDX     #$0085  CHRCOD FOR
PROVISIONS
6866 BF   67E5     01080          STX     CHRCOD
6869 CC   0005     01090          LDD     #5      MAXIMUM MY
686C BD   5A60     01100          JSR     RANDOM  RANDOM #
GENERATOR
686F F7   67E4     01110          STB     YCOORD  MY-COORDINATE
```

```
6872 C1   00        01120         CMPB    #0      IS IT FIRST
ROW?
6874 27   0C        01130         BEQ     LBL008  GO IF YES
6876 C1   05        01140         CMPB    #5      IS IT LAST ROW?
6878 27   13        01150         BEQ     LBL009  GO IF YES
687A CC   000E      01160         LDD     #14     RANDOM RANGE
ROWS 1-4
687D BD   5A60      01170         JSR     RANDOM  RANDOM #
GENERATOR
6880 20   11        01180         BRA     LBL010
6882 CC   0009      01190 LBL008  LDD     #9      RANDOM RANGE
ROW 0
6885 BD   5A60      01200         JSR     RANDOM  RANDOM #
GENERATOR
6888 C3   0005      01210         ADDD    #5      OFFSET FOR ROW
0
688B 20   06        01220         BRA     LBL010
688D CC   0009      01230 LBL009  LDD     #9      RANDOM RANGE
ROW 5
6890 BD   5A60      01240         JSR     RANDOM  RANDOM #
GENERATOR
6893 F7   67E3      01250 LBL010  STB     XCOORD  MX-COORDINATE
6896 B6   67E3      01260 LBL011  LDA     XCOORD  MX-COORDINATE
6899 F6   67E4      01270         LDB     YCOORD  MY-COORDINATE
689C BD   5A80      01280         JSR     MCSCCV  CONVERT TO
(SX,SY)
689F BD   5B80      01290         JSR     GTFVAL  GET CURRENT
CONTENTS
68A2 C1   20        01300         CMPB    #$20    IS IT A BLANK
SPACE?
68A4 27   39        01310         BEQ     LBL015  GO IF YES
68A6 B6   67E3      01320         LDA     XCOORD  MX-COORDINATE
68A9 F6   67E4      01330         LDB     YCOORD  MY-COORDINATE
68AC 4C             01340         INCA            INCREMENT MX
68AD B7   67E3      01350         STA     XCOORD  SAVE IT
68B0 C1   05        01360         CMPB    #5      IS IT LAST ROW?
68B2 26   07        01370         BNE     LBL012  GO IF NO
68B4 C6   0A        01380         LDB     #10     LAST ROW RANGE
68B6 F7   67E7      01390         STB     RANGE
68B9 20   02        01400         BRA     LBL013
68BB C6   0F        01410 LBL012  LDB     #15     OTHER ROWS
RANGE
68BD B1   67E7      01420 LBL013  CMPA    RANGE   END OF ROW?
68C0 26   D4        01430         BNE     LBL011  CHECK NEXT CELL
IF NO
68C2 F6   67E4      01440         LDB     YCOORD  MY-COORDINATE
68C5 5C             01450         INCB            INCREMENT MY
```

```
68C6 F7   67E4      01460          STB     YCOORD  SAVE IT
68C9 C1   06        01470          CMPB    #6      END OF SCREEN
68CB 27   07        01480          BEQ     LBL014  GO IF YES
68CD 86   00        01490          LDA     #0      MX-COORDINATE
68CF B7   67E3      01500          STA     XCOORD  SAVE IT
68D2 20   C2        01510          BRA     LBL011  GO CHECK NEXT
CELL
68D4 86   05        01520 LBL014   LDA     #5      MX-COORDINATE
68D6 B7   67E3      01530          STA     XCOORD  SAVE IT
68D9 5F             01540          CLRB            MY-COORDINATE
68DA F7   67E4      01550          STB     YCOORD  SAVE IT (=0)
68DD 20   B7        01560          BRA     LBL011  GO CHECK NEXT
CELL
68DF B6   67E3      01570 LBL015   LDA     XCOORD  MX-COORDINATE
68E2 F6   67E4      01580          LDB     YCOORD  MY-COORDINATE
68E5 BD   5A80      01590          JSR     MCSCCV  CONVERT TO
(SX,SY)
68E8 BE   67E5      01600          LDX     CHRCOD  CHARACTER CODE
68EB BD   5B60      01610          JSR     PTFVAL  PUT TO SCREEN
BUFFER
                    01620
                    01630 * INITIALIZE THE SCREEN
68EE 108E 5500      01640          LDY     #LINE00 POINT TO
BUFFERS
                    01650
68F2 86   FF        01660          LDA     #$FF    SET FIRST
XCOORD TO ROLL
68F4 B7   67E3      01670          STA     XCOORD
68F7 C6   00        01680          LDB     #$00    SET FIRST
YCOORD TO ZERO
68F9 F7   67E4      01690          STB     YCOORD
                    01700
68FC B6   67E3      01710 LBL002   LDA     XCOORD
68FF F6   67E4      01720          LDB     YCOORD
6902 4C             01730          INCA            INCREMENT
XCOORD
6903 B7   67E3      01740          STA     XCOORD
6906 F7   67E4      01750          STB     YCOORD
6909 81   20        01760          CMPA    #32     END OF THE X
LINE?
690B 25   0E        01770          BLO     LBL003  GO IF NO
690D 4F             01780          CLRA            SET XCOORD = 0
690E B7   67E3      01790          STA     XCOORD
6911 5C             01800          INCB            INCREMENT
YCOORD
6912 F7   67E4      01810          STB     YCOORD
6915 C1   10        01820          CMPB    #16     END OF SCREEN?
```

```
6917 25   02        01830         BLO     LBL003  GO IF NO
6919 20   22        01840         BRA     LBL004  GO IF YES
                    01850
691B B6   67E3      01860 LBL003  LDA     XCOORD
691E F6   67E4      01870         LDB     YCOORD
6921 BD   5B80      01880         JSR     GTFVAL  GET CELL
CONTENTS
6924 4F             01890         CLRA            EXTEND IT
6925 1F   01        01900         TFR     D,X     MOVE IT TO REG
X
6927 BF   67E5      01910         STX     CHRCOD  SAVE IT
                    01920
692A 34   16        01930         PSHS    A,B,X   PUT CHRCOD TO
SCREEN
692C B6   67E3      01940         LDA     XCOORD
692F F6   67E4      01950         LDB     YCOORD
6932 BE   67E5      01960         LDX     CHRCOD
6935 BD   5300      01970         JSR     PTFCHR
6938 35   16        01980         PULS    A,B,X
693A 16   FFBF      01990         LBRA    LBL002  RETURN FOR NEXT
CHRCOD
                    02000
                    02010 * INITIAL AVATAR SETUP
693D BD   5AE0      02020 LBL004  JSR     STAVTR
                    02030
                    02040 * REPORT CURRENT STRENGTH
6940 BD   6400      02050         JSR     RPTSTR
                    02060
                    02070 * REPORT CURRENT SCORE
6943 BD   6460      02080         JSR     RPTSCO
                    02090
                    02100 * GAME LOOP
6946 BD   6960      02110         JSR     GMLOOP
                    02120
                    02130 * HOLD THE SCREEN
6949 20   FE        02140 LBL016  BRA     LBL016
                    02150
694B 35   76        02160         PULS    A,B,X,Y,U
694D 39             02170 ENDCHK  RTS
                    02180
          0000      02190         END
```

=====

GMLOOP: The Game Loop

The Assembly Language text listing:

```
                   *****
                   *
                   * GMLOOP.ASM
                   * MDJ 2024/02/13
                   *
                   * GAME LOOP
                   *
                   *****

                   * MLF POLCAT
          4142     POLCAT  EQU     $4142

                   * KEY EVENT HANDLERS
          5CA0     BCHARK  EQU     $5CA0
          5CC0     DCHARK  EQU     $5CC0
          5CE0     EASTK   EQU     $5CE0
          5DE0     GCHARK  EQU     $5DE0
          5E20     ICHARK  EQU     $5E20
          5E40     LCHARK  EQU     $5E40
          5F00     NCHARK  EQU     $5F00
          5F20     NORTHK  EQU     $5F20
          5FE0     PCHARK  EQU     $5FE0
          6000     RCHARK  EQU     $6000
          6020     SOUTHK  EQU     $6020
          60E0     TCHARK  EQU     $60E0
          6160     UCHARK  EQU     $6160
          6180     WESTK   EQU     $6180
          6240     XCHARK  EQU     $6240
          6280     YCHARK  EQU     $6280

6960                       ORG     $6960

6960 34   02       GMLOOP  PSHS    A

                   * GET A KEYPRESS
6962 BD   4142     LBL001  JSR     POLCAT
6965 27   FB               BEQ     LBL001

                   * BCHARK (B-KEY = BAG INVENTORY)
6967 81   42       LBL002  CMPA    #$42
6969 26   05               BNE     LBL003
696B BD   5CA0             JSR     BCHARK
```

```
696E 20   F2        00520         BRA     LBL001
                    00530
                    00540 * DCHARK (D-KEY = DOWN TO NEXT LEVEL
BELOW)
                    00550 * (UNIMPLEMENTED DUMMY)
6970 81   44        00560 LBL003  CMPA    #$44
6972 26   05        00570         BNE     LBL004
6974 BD   5CC0      00580         JSR     DCHARK
6977 20   E9        00590         BRA     LBL001
                    00600
                    00610 * EASTK (RIGHT ARROW)
6979 81   09        00620 LBL004  CMPA    #$09
697B 26   05        00630         BNE     LBL005
697D BD   5CE0      00640         JSR     EASTK
6980 20   E0        00650         BRA     LBL001
                    00660
                    00670 * GCHARK (G-KEY = NEW GAME)
6982 81   47        00680 LBL005  CMPA    #$47
6984 26   05        00690         BNE     LBL006
6986 BD   5DE0      00700         JSR     GCHARK  JUMPS TO SMREAD
6989 20   D7        00710         BRA     LBL001  DUMMY
                    00720
                    00730 * ICHARK (I-KEY = WAREHOUSE INVENTORY)
698B 81   49        00740 LBL006  CMPA    #$49
698D 26   05        00750         BNE     LBL007
698F BD   5E20      00760         JSR     ICHARK
6992 20   CE        00770         BRA     LBL001
                    00780
                    00790 * LCHARK (L-KEY = LEAVE)
                    00800 * (EMPTY CONTENTS OF BAG INTO CELL)
6994 81   4C        00810 LBL007  CMPA    #$4C
6996 26   05        00820         BNE     LBL008
6998 BD   5E40      00830         JSR     LCHARK
699B 20   C5        00840         BRA     LBL001
                    00850
                    00860 * NCHARK (N-KEY = ANSWER "NO")
                    00870 * (DO NOT CONFIRM)
                    00880 * (UNIMPLEMENTED DUMMY)
699D 81   4E        00890 LBL008  CMPA    #$4E
699F 26   05        00900         BNE     LBL009
69A1 BD   5F00      00910         JSR     NCHARK
69A4 20   BC        00920         BRA     LBL001
                    00930
                    00940 * NORTHK (UP ARROW)
69A6 81   5E        00950 LBL009  CMPA    #$5E
69A8 26   05        00960         BNE     LBL010
69AA BD   5F20      00970         JSR     NORTHK
```

```
69AD 20   B3                 BRA     LBL001

                 * PCHARK (P-KEY = PAUSE GAME)
                 * (UNIMPLEMENTED DUMMY)
69AF 81   50     LBL010  CMPA    #$50
69B1 26   05             BNE     LBL011
69B3 BD   5FE0           JSR     PCHARK
69B6 20   AA             BRA     LBL001

                 * RCHARK (R-KEY = RESUME GAME)
                 * (UNIMPLEMENTED DUMMY)
69B8 81   52     LBL011  CMPA    #$52
69BA 26   05             BNE     LBL012
69BC BD   6000           JSR     RCHARK
69BF 20   A1             BRA     LBL001

                 * SOUTHK (DOWN ARROW)
69C1 81   0A     LBL012  CMPA    #$0A
69C3 26   05             BNE     LBL013
69C5 BD   6020           JSR     SOUTHK
69C8 20   98             BRA     LBL001

                 * TCHARK (T-KEY = TAKE)
                 * (PUT CONTENTS OF CELL INTO BAG)
69CA 81   54     LBL013  CMPA    #$54
69CC 26   05             BNE     LBL014
69CE BD   60E0           JSR     TCHARK
69D1 20   8F             BRA     LBL001

                 * UCHARK (U-KEY = UP TO NEXT LEVEL
ABOVE)
                 * (UNIMPLEMENTED DUMMY)
69D3 81   55     LBL014  CMPA    #$55
69D5 26   05             BNE     LBL015
69D7 BD   6160           JSR     UCHARK
69DA 20   86             BRA     LBL001

                 * WESTK (LEFT ARROW)
69DC 81   08     LBL015  CMPA    #$08
69DE 26   06             BNE     LBL016
69E0 BD   6180           JSR     WESTK
69E3 16   FF7C           LBRA    LBL001

                 * XCHARK (X-KEY = EXIT GAME)
69E6 81   58     LBL016  CMPA    #$58
69E8 26   06             BNE     LBL017
69EA BD   6240           JSR     XCHARK  JUMPS TO COLD
```

```
69ED 16   FF72             LBRA    LBL001  DUMMY

                   * YCHARK (Y-KEY = ANSWER "YES")
                   * (CONFIRM)
                   * (UNIMPLEMENTED DUMMY)
69F0 81   59       LBL017  CMPA    #$59
69F2 26   06               BNE     LBL018
69F4 BD   6280             JSR     YCHARK
69F7 16   FF68             LBRA    LBL001

                   * ANY OTHER KEYPRESS
69FA 16   FF65     LBL018  LBRA    LBL001

69FD 35   02       GMEXIT  PULS    A
69FF 39            ENDCHK  RTS

          0000             END
```

=====

MALKYS.BAS: Sets General Parameters, enters ALLRAM Mode, and then Executes the SMREAD Routine Which jumps to the SMGAME Routine

The BASIC Language program listing:

```
1000 '*****
1010 '*
1020 '* MALKYS.BAS
1030 '* MDJ 2024/02/13
1040 '*
1050 '* MALKY'S WARREN: THE
1060 '* FIRST TRAINING QUEST
1070 '*
1080 '* SCREEN MAZE GAME
1090 '* BASIC PROGRAM
1100 '*
1110 '* DISPLAYS THE MAZE
1120 '* ON SCREEN, USING THE
1130 '* FAKETEXT 32 X 16
1140 '* CHARACTER SET FOR
1150 '* PMODE 4, AND THEN
1160 '* STARTS THE GAME.
1170 '*
1180 '*****
1190 '

1500 PRINT
1510 PRINT "WORKING ***";

2000 'SETUP MEMORY
2010 CLEAR 0,&H4000
2020 PCLEAR 4
2030 PRINT "***";
2040 '

4000 LOADM "MALKYS.BIN"
4010 PRINT "***";
4020 '

5000 'SETUP GRAPHICS
5010 PMODE 4,1
```

```
5020 PCLS 1
5030 SCREEN 1,0
5040 '

7000 'SMREAD.ASM RUN ADDRESS = &H5AA0
7005 '(HEREIN, SMREAD JUMPS TO SMGAME)
7010 'PUT IT TO THE ML FOUNDATION'S
7020 'REGPC (AT $H400A)
7030 POKE &H400A, &H5A
7040 POKE &H400B, &HA0
7050 'GO START THE GAME IN ALLRAM MODE
7060 EXEC &H4403  'STRTUP
7070 '

32767 END
```

=====

Results

Well, the Warren works. The game is playable and does indeed serve as the Proof-of-Concept it was primarily intended to be.

=====

Conclusions and Future Work

In addition to correction of the “CoCo 3 only” problem and the “ X “ Key bug, the following are areas to be addressed in future work:

1. SMDISPLY.ASM is somewhat inefficient, but it is only used for examining mazes which have already been built. It is still much faster than the user, so no further work on this routine is contemplated.

2. DECMAL is a bit Brute Force-ish. I may want to investigate the possibility of doing something more elegant. However, I suspect that the Brute Force method may actually be the most efficient here.

3. PTFSLS.ASM is a highly abbreviated length-specified fake text print string mechanism. It should only be used for maze games. A more general routine could be developed, but I see little use for such at the moment.

4. RPTSTR.ASM and RPTSCO.ASM both use brute-force methodology, but such is considered to be most efficient here.

5. The False Disk System should yield itself to the creation of mazes with many Levels and Sections; and the Fake Text System was specifically designed to provide significant possibilities beyond the simple One Level, One Section scheme of Malky’s Warren. Many mazes await to be designed. Perhaps partially random maze generation may be considered.

6. The next project to be addressed is a Second Training Quest, using a more complex combination of maze levels and sections; and including more items to be retrieved and more provisions to be assimilated. This will be intended as a middle step between Malky’s Warren and the truly huge maze systems this methodology should be able to accommodate.

=====

MALKY’S WARREN
Cheat Sheet
Version 1.0.1

General Guidelines:

1. To start the Quest, put the MALKYS.DSK into Drive 0 and enter RUN “MALKYS.BAS”.

2. The moment you exit the Warren, the game is over. There’s no going back at that point. Be careful not to go East from the Warehouse (marked “W”) accidentally.

3. North is up on the screen. Press the “ Up-Arrow ” to go North. Press the “ Right-Arrow ” to go East. Press the “ Down-Arrow ” to go South. Press the “ Left-Arrow ” to go West.

4. Press the “ T ” Key to Take something and put it in your Bag. Press the “ L ” Key to take something out of your Bag and Leave it in the Current Cell (including the Warehouse Cell).

5. Press the “ B ” Key for a Bag Contents List. Press the “ I ” Key for a Warehouse Inventory List.

6. Press the “G” Key for a New Game. Press the “X” Key to Exit back to CoCo 2 Disk Basic.

______

Key Codes

B - Bag Inventory Report

G - New Game?

I - Warehouse Inventory Report

L - Leave (Empty the Bag)

N - “No” (Do Not Confirm)

T - Take (Put to Bag)

X Exit Game?

Y - “Yes” (Confirm)

Up Arrow - Go North

Right Arrow - Go East

Down Arrow - Go South

Left Arrow - Go West

=====

Appendix A
Decimal to Hexadecimal Conversions

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
000	00	032	20	064	40	096	60
001	01	033	21	065	41	097	61
002	02	034	22	066	42	098	62
003	03	035	23	067	43	099	63
004	04	036	24	068	44	100	64
005	05	037	25	069	45	101	65
006	06	038	26	070	46	102	66
007	07	039	27	071	47	103	67
008	08	040	28	072	48	104	68
009	09	041	29	073	49	105	69
010	0A	042	2A	074	4A	106	6A
011	0B	043	2B	075	4B	107	6B
012	0C	044	2C	076	4C	108	6C
013	0D	045	2D	077	4D	109	6D
014	0E	046	2E	078	4E	110	6E
015	0F	047	2F	079	4F	111	6F
016	10	048	30	080	50	112	70
017	11	049	31	081	51	113	71
018	12	050	32	082	52	114	72
019	13	051	33	083	53	115	73
020	14	052	34	084	54	116	74
021	15	053	35	085	55	117	75
022	16	054	36	086	56	118	76
023	17	055	37	087	57	119	77
024	18	056	38	088	58	120	78
025	19	057	39	089	59	121	79
026	1A	058	3A	090	5A	122	7A
027	1B	059	3B	091	5B	123	7B
028	1C	060	3C	092	5C	124	7C
029	1D	061	3D	093	5D	125	7D
030	1E	062	3E	094	5E	126	7E
031	1F	063	3F	095	5F	127	7F

DEC	HEX	DEC	HEX	DEC	HEX	DEC	HEX
128	**80**	**160**	**A0**	**192**	**C0**	**224**	**E0**
129	**81**	**161**	**A1**	**193**	**C1**	**225**	**E1**
130	**82**	**162**	**A2**	**194**	**C2**	**226**	**E2**
131	**83**	**163**	**A3**	**195**	**C3**	**227**	**E3**
132	**84**	**164**	**A4**	**196**	**C4**	**228**	**E4**
133	**85**	**165**	**A5**	**197**	**C5**	**229**	**E5**
134	**86**	**166**	**A6**	**198**	**C6**	**230**	**E6**
135	**87**	**167**	**A7**	**199**	**C7**	**231**	**E7**
136	**88**	**168**	**A8**	**200**	**C8**	**232**	**E8**
137	**89**	**169**	**A9**	**201**	**C9**	**233**	**E9**
138	**8A**	**170**	**AA**	**202**	**CA**	**234**	**EA**
139	**8B**	**171**	**AB**	**203**	**CB**	**235**	**EB**
140	**8C**	**172**	**AC**	**204**	**CC**	**236**	**EC**
141	**8D**	**173**	**AD**	**205**	**CD**	**237**	**ED**
142	**8E**	**174**	**AE**	**206**	**CE**	**238**	**EE**
143	**8F**	**175**	**AF**	**207**	**CF**	**239**	**EF**
144	**90**	**176**	**B0**	**208**	**D0**	**240**	**F0**
145	**91**	**177**	**B1**	**209**	**D1**	**241**	**F1**
146	**92**	**178**	**B2**	**210**	**D2**	**242**	**F2**
147	**93**	**179**	**B3**	**211**	**D3**	**243**	**F3**
148	**94**	**180**	**B4**	**212**	**D4**	**244**	**F4**
149	**95**	**181**	**B5**	**213**	**D5**	**245**	**F5**
150	**96**	**182**	**B6**	**214**	**D6**	**246**	**F6**
151	**97**	**183**	**B7**	**215**	**D7**	**247**	**F7**
152	**98**	**184**	**B8**	**216**	**D8**	**248**	**F8**
153	**99**	**185**	**B9**	**217**	**D9**	**249**	**F9**
154	**9A**	**186**	**BA**	**218**	**DA**	**250**	**FA**
155	**9B**	**187**	**BB**	**219**	**DB**	**251**	**FB**
156	**9C**	**188**	**BC**	**220**	**DC**	**252**	**FC**
157	**9D**	**189**	**BD**	**221**	**DD**	**253**	**FD**
158	**9E**	**190**	**BE**	**222**	**DE**	**254**	**FE**
159	**9F**	**191**	**BF**	**223**	**DF**	**255**	**FF**

=====

Appendix B: My CoCo Philosophy

The CoCo community enjoys a great diversity of interests.

Some choose to concentrate on hardware innovations and modifications such as interfacing with VGA and HDMI monitors, SD Card data storage, and 104-key keyboards. This interest is at least partly born of necessity, since composite monitors, floppy diskettes, and CoCo spare parts are no longer manufactured and are in increasingly short supply.

Others concentrate on expanding the software horizons of the CoCo 3, using NitrOS-9 and other operating systems to make the multitasking CoCo behave ever closer to modern Windows, Mac, and Linux machines.

Still others are devoted to emulating the CoCo on other platforms by developing emulators such as VCC, OVCC, MAME, and XRoar.

And some just love retro gaming.

My personal interest is twofold:

1. To see VCC increasingly used as a learning tool for budding software developers.

2. To see just how much I can cram into a 64K CoCo 2.

First, VCC: Today's Grade School, Junior High, and High School students have a wealth of available learning tools. Micro-bits, Arduinos, and Raspberry Pi supermicro devices provide highly affordable entry-level introductions to computer programming and interfacing. Maker-Spaces and Innovation Centers in our schools and libraries help foster growth and experience.

But these devices do have limitations. Even these simple(?) computers can have rather steep learning curves, and their low initial cost can quickly expand as new peripherals and experimental equipment and supplies are added.

VCC is free, and can be used on any Windows computer: just download it, install it, and it runs. If you don't own a Windows computer, your school, library, or a friend probably does. The included BASIC language is easy to learn and can readily serve as a stepping-stone towards more complex programming languages. (And, no, learning structured programming does not require a language that enforces structure. In fact, I think learning to structure your programs is actually more effective when you do so on your own.)

I prefer VCC to the other emulators for these purposes because its setup is trivial: Again, just download it, install it, and it runs. OVCC, MAME, and XRoar have their advantages, but ease of setup is not one of them. Even with their available Windows binary packages, they require pre-installation of other bits and pieces of software before they can be downloaded, installed, and run. This may not be a major problem for a reasonably adept aficionado, but it forms a significant barrier for the newbie. And, it’s the newbie whom we’re trying to reach, interest, and encourage here; the newbie who may not yet recognize even the tiniest awakening of interest in things computational.

But, for these purposes, VCC has one glaring weakness: its instruction manual is woefully terse. I would like to see VCC bundled with a selection of tutorials, manuals, and examples suited to guiding even the most newbie of newbies into the wonders of computing.

Second, The Stuffed CoCo: I’m simply fascinated by the challenge of seeing how much functional capability I can sandwich into the nooks and crannies of the 64K space. Whether it’s working in the available RAM left by the 32K ROM and the dedicated RAM that supports that ROM, or whether it’s jumping right into ALLRAM mode and just filling the entire 64K to near-overflowing; it’s an investigative gauntlet which goes right to the heart of my enchantment with puzzles in general.

It’s great fun!

M.D.J. 2021/08/29

=====

Appendix C: New BDS Software License

This New Software License applies to all software found on the BDS Software site, and supersedes all previous copyright notices and licensing provisions which may appear in the software itself or in any documentation therefor.

All software which has previously been placed in the public domain remains in the public domain.

All other software, programs, experiments and reports, documentation, and any other material on this site (other than that attributed to outside sources) is hereby copyright © 2018 (or later if so marked) by M. David Johnson.

All software, documentation, and other information on the BDS Software site is available for you to freely download without cost.

Whether you downloaded such items directly from this site, or you obtained them by any other means, you are hereby licensed to copy them, to sell or give away such copies, to use them, and to excerpt from them, in any way whatsoever, so long as nothing you do with them would denigrate the name of our Lord and Savior, Jesus Christ.

I make absolutely no warranty whatsoever for any of these items. You use them entirely at your own risk.

If they don't work for you, I commiserate.

If they crash your system, I sympathize.

But I accept no responsibility whatsoever for any such consequences. Under no circumstances will BDS Software or M. David Johnson be liable for any negative results of any kind which you may experience from downloading or using these items.

BDS Software's former mail address at P.O. Box 485 in Glenview, IL is no longer valid. Any mail sent to that address will be rejected by the U.S. Postal Service. See my Contact page.

M.D.J. 2018/06/08

=====

Works Cited

Wikipedia “Diocletianic Persecution”.
https://en.wikipedia.org/wiki/Diocletianic_Persecution . 2023. Online.

=====