A Guide to Creating Your Own

TUNNELS OF DOOM

Adventure

On the Texas Instruments Home Computer

A Guide to Creating Your Own Tunnels of Doom Adventure On the Texas Instruments Home Computer

By **Michael Veprauskas**

This Guide is dedicated to the extensive legacy in Adventure gaming made possible by **Texas Instruments, Inc.** through its development and production of the Texas Instruments Home Computer

and

to all true Adventurers at heart.!

Table of Contents

Introduction

Tunnels of Doom made its début almost 40 years ago in 1982. At that time it was the only true D&D type game option available to individuals who owned a TI99/4 or TI99/4a console and were limited to a cassette recorder based storage system. Back in 1982 the vast majority of TI Home Computer owners fell into that category. In addition, ***Tunnels of Doom*** was a rather pricey package. The suggested retail price for the combined Game Module and database package was $59.95, making it the most expensive "Entertainment" category Command Module yet released. The Command Module was bundled with two game databases, which were available on either cassette or diskette: ***Quest for the King*** and ***Pennies and Prizes***. A 40 page informative instructional booklet completed the package.

The ***Tunnels of Doom*** Command Module added a whopping "30K bytes of active memory" or GROM to the console system. Some years ago the author of the port of the TOD Module to an Editor/ Assembler disk based platform recalled his relief that this could be done, "I had something like 10 bytes of free memory left when it was all finished, but it worked!" In addition to the module GROM that contained the program proper, the 16K console VDP memory was utilized to store the game database. Whether a game database is loaded to VDP memory from diskette, or saved from VDP memory to diskette, each is a faithful reflection of the other. In fact it is such a reliable dumping of VDP memory that early on, owners of cassette based ***Adventure Module*** programs used the TOD module to convert their programs to diskette media! However, there were those occasional odd characters….

The ability to load individual game databases to VDP memory was a promise that new and unique games could, and would be forthcoming in the near future. With the demise of the Texas Instruments Home Computer Division, this future development was left to the end users for fulfillment. That this eventually became possible was due to the development of Sector Editors and the related ***Tunnels of Doom*** Editors.

Two existing ***Tunnels of Doom*** Editors are in current use. The first dates from 1985 and was the medium through which 90% (plus) of all ***Tunnels of Doom*** game databases were designed. This editor was written by John Behnke and distributed by Asgard Software.

During the later 1980's I acquired the ***Tunnels of Doom Editor*** written by Behnke. I had previously acquired some familiarity with the use of a Sector Editor for modifying software, in particular for configuration of printers and storage options. It was obvious that the Asgard Editor was in fact a modified Sector Editor, tasked with the specific objective of editing a ***Tunnels of Doom*** game database. Several months of enjoyable experimentation followed, during which several new game databases were designed. This Editor, although extremely useful had its limitations.

Probably the most significant limitation was the Editor's inability to modify all 56 Monster profiles, which are universally, though randomly, accessed in a ten-level dungeon. This produced the unintended and quite unwanted effect of Monsters designed for ***Quest*** appearing out of nowhere, but never in a predictable manner, in a new game design. An inspection with a sector editor revealed that there were in fact a total of 56 Monster profiles that required editing. The very effort required to determine this fact led to other discoveries.

It was also noted that the ***Pennies*** database was in fact an edited and modified ***Quest*** database. Remnants of many defined graphics from ***Quest*** in particular 3 sections of Vault graphics, which have no purpose in ***Pennies***, are found scattered therein. The inevitable question occurred, "how do ***Quest*** and ***Pennies*** access completely different graphics and colors, for different purposes, in different memory locations?" The **Asgard Editor** only permitted redefinition of existing graphics, using the identical sizes and colors of those redefined; ***Pennies*** demonstrated that somewhat more was possible. These questions required solutions. The end result of this line of enquiry was ***Halls of Lost… Moria!***, which was completed toward the end of 1993 and redesigned in 1994. The original graphics for ***Halls*** were created in a 9958 video processor environment; these simply looked terrible and were frequently blurred when viewed with the original 99/4a console video chip. The revised program was distributed by *RamCharged Computers* the following year. Unfortunately, however, ***Halls*** appeared during the very terminal phase of the TI99/4a Home Computer as a computer of first choice. In addition, life intervened.

A second Tunnels of Doom Editor was released in 2015 by Fritzling Software. Fritz's software represents a quantum improvement in TOD game database design. The old Asgard Editor is sector based; meaning that it presumes the game base begins on **Byte 00** (in hex) of **Sector 0022**, extending through **Sector 0054** and modifies selected editable parameters accordingly. To accommodate these assumptions the game database must be the first program or file on a diskette, real or image based, for the editing to succeed. Fritz's Editor operates from a PC platform, both Editor and database can be located anywhere on any of your computer's permanent or removable drives. The editing is not sector and byte oriented, but proceeds from the first database game byte to the last. These features greatly increase design flexibility. In addition, support files include both sound clips to assist in selecting monster sounds and a TOD oriented graphics collection. It is hoped that the information provided here will inspire an updated version to that effort. Given the number of editable options in a game database it is unlikely that a TI based Editor, due to the TI99/4a's memory limitations, would ever reasonably suffice. Theoretically, a PC based TOD editor would have no such limitations.

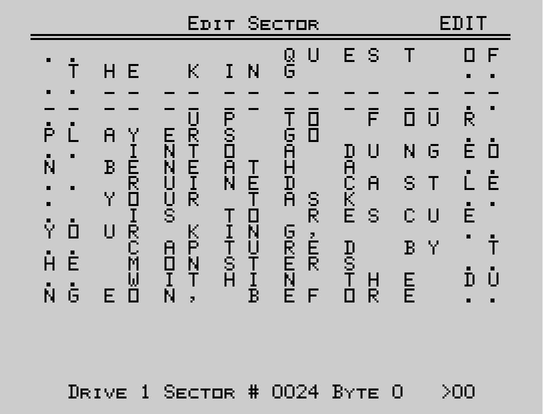
This booklet was written with game development in mind. Each section or chapter focuses on a manageable portion, hopefully complete in itself. Get started immediately on a game <u>theme</u> and <u>graphic</u> designs to visually illustrate your idea. Graphic design will prove to be the most time consuming element for your database. At least it should be!!!

The final three chapters on *Dungeon Design, Global Options and Game Scenarios* are considered advanced topics and might reasonably be deferred until some practical experience with the previous material has been achieved. Compilation of a personal database for graphics: hallways, room items, monsters, etc. and the various "lists" are highly desirable. As noted Fritz's Editor comes complete with such a compilation and forms an excellent starting point.

Game Creation Options

Designing a new TOD database will inevitably require the modification, or editing, of an existing game database. These can all be traced back to the original ***Quest for the King*** database created by Kevin Kenney of ***Texas Instruments***. Currently there are three platforms from which you may elect to modify an existing database: using TI99/4a or compatible hardware, a PC based TI99/4a emulator, or a PC based Hex Editor.

When using original TI99/4a or compatible hardware (TI99/4, Geneve, etc.) editing is performed with TI software capable of editing TI formatted diskettes. These Sector Editors use a Sector/ Byte reference framework to identify the disk memory locations for editing. To facilitate this approach the database is optimally the first file on the diskette so that a Sector/ Byte reference guide may be employed for the necessary modifications. A reference guide of this sort is the stated purpose of this booklet. If the game database developer opts to place the database elsewhere on the diskette, the references in this guide will need to be adjusted accordingly. However, it is far simpler to place the database as the first file on the disk and commence at **Sector 0022**. Editing options using this approach include the original ***Asgard Editor***, ***DSKU*** and similar T.I. Sector Editors.

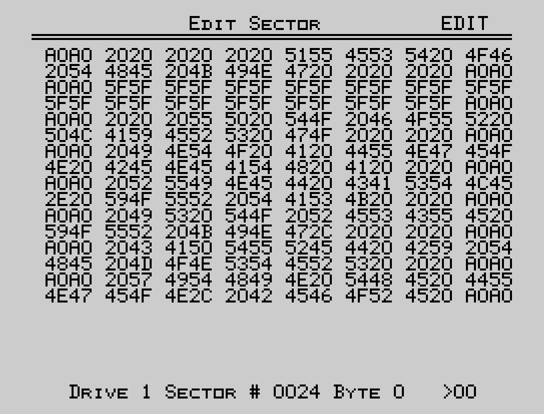


```
                 Edit Sector              EDIT

. .                 Q U  E S  T    O F
  T  H E    K  I N  G                 . .
. .  - -  - -  - -  - -  - -  - -  - -
- -  - -  - -  - -  - -  - -  - -  . .
. .       U  P     T O    F  O U  R
P L  A Y  E R  S    G O            . .
. .    I  N T  O    A    D U  N G  E O
N    B E  N E  A T  H    A         . .
. .    R  U I  N E  D    C A  S T  L E
.    Y O  U R    T  A S  K         . .
. .    I  S    T O    R  E S  C U  E
Y O  U R    K  I N  G ,            . .
. .    C  A P  T U  R E  D    B Y    T
H E    M  O N  S T  E R  S         . .
. .    W  I T  H I  N    T H  E    D U
N G  E O  N ,    B  E F  O R  E    . .

  Drive 1 Sector # 0024 Byte 0    >00
```

Sector 0024 of the Quest Database (DSKU)



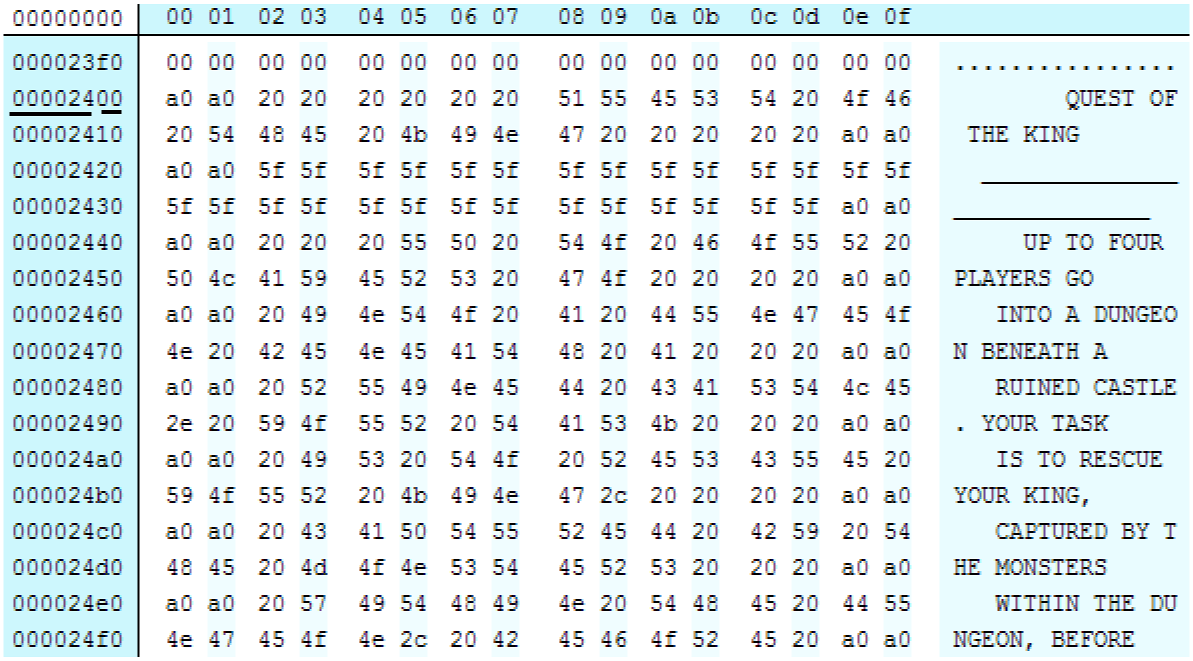
```
                 Edit Sector              EDIT

A0A0 2020 2020 2020 5155 4553 5420 4F46
2054 4845 204B 494E 4720 2020 2020 A0A0
A0A0 5F5F 5F5F 5F5F 5F5F 5F5F 5F5F 5F5F
5F5F 5F5F 5F5F 5F5F 5F5F 5F5F 5F5F A0A0
A0A0 2020 2055 5020 544F 2046 4F55 5220
504C 4159 4552 5320 474F 2020 2020 A0A0
A0A0 2049 4E54 4F20 4120 4455 4E47 454F
4E20 4245 4E45 4154 4820 4120 2020 A0A0
A0A0 2052 5549 4E45 4420 4341 5354 4C45
2E20 594F 5552 2054 4153 4B20 2020 A0A0
A0A0 2049 5320 544F 2052 4553 4355 4520
594F 5552 204B 494E 472C 2020 2020 A0A0
A0A0 2043 4150 5455 5245 4420 4259 2054
4845 204D 4F4E 5354 4552 5320 2020 A0A0
A0A0 2057 4954 4849 4E20 5448 4520 4455
4E47 454F 4E2C 2042 4546 4F52 4520 A0A0

  Drive 1 Sector # 0024 Byte 0    >00
```

Same Sector in Hexadecimal Notation

With the development of PC based TI99/4a emulators and their associated file and “disk image” software a new option became available for ***TOD*** database modifications. It is now possible to edit ‘virtual disks’, or ‘disk images’ located on your PC. Optimally these disk images represent a byte by byte duplication of a physical diskette. These disk images can be created on a PC using software such as ***Classic99***, ***TIDir***, or some similar utility. The database may then be edited with software running on the emulator in precisely the same manner that occurs with a physical diskette, for the methods used to edit physical diskette media and compatible disk images are the same. The database should be the first file on the disk image (i.e. virtual **Sector 0022**) and editing location references are by Sector and Byte. The same TI99/4a software usable for this purpose may also be used with an emulator. If you choose to use ***DSKU*** for your editor, be certain to obtain version 4.0 of the software. Version 4.0 will correctly indicate the Sector/ Byte location of your editing cursor; earlier versions appear to provide only the Sector location.



```
00000000 | 00 01  02 03  04 05  06 07  08 09  0a 0b  0c 0d  0e 0f
---------+--------------------------------------------------------------------------
000023f0 | 00 00  00 00  00 00  00 00  00 00  00 00  00 00  00 00   ................
00002400 | a0 a0  20 20  20 20  20 20  51 55  45 53  54 20  4f 46           QUEST OF
00002410 | 20 54  48 45  20 4b  49 4e  47 20  20 20  20 20  a0 a0    THE KING
00002420 | a0 a0  5f 5f  5f 5f  5f 5f  5f 5f  5f 5f  5f 5f  5f 5f     ______________
00002430 | 5f 5f  5f 5f  5f 5f  5f 5f  5f 5f  5f 5f  5f 5f  a0 a0   ______________
00002440 | a0 a0  20 20  20 55  50 20  54 4f  20 46  4f 55  52 20        UP TO FOUR
00002450 | 50 4c  41 59  45 52  53 20  47 4f  20 20  20 20  a0 a0   PLAYERS GO
00002460 | a0 a0  20 49  4e 54  4f 20  41 20  44 55  4e 47  45 4f      INTO A DUNGEO
00002470 | 4e 20  42 45  4e 45  41 54  48 20  41 20  20 20  a0 a0   N BENEATH A
00002480 | a0 a0  20 52  55 49  4e 45  44 20  43 41  53 54  4c 45      RUINED CASTLE
00002490 | 2e 20  59 4f  55 52  20 54  41 53  4b 20  20 20  a0 a0   . YOUR TASK
000024a0 | a0 a0  20 49  53 20  54 4f  20 52  45 53  43 55  45 20      IS TO RESCUE
000024b0 | 59 4f  55 52  20 4b  49 4e  47 2c  20 20  20 20  a0 a0   YOUR KING,
000024c0 | a0 a0  20 43  41 50  54 55  52 45  44 20  42 59  20 54      CAPTURED BY T
000024d0 | 48 45  20 4d  4f 4e  53 54  45 52  53 20  20 20  a0 a0   HE MONSTERS
000024e0 | a0 a0  20 57  49 54  48 49  4e 20  54 48  45 20  44 55      WITHIN THE DU
000024f0 | 4e 47  45 4f  4e 2c  20 42  45 46  4f 52  45 20  a0 a0   NGEON, BEFORE
```

Sector 0024 of the Quest Database (Disk Image) Viewed with Hex Editor Neo

The Sector <u>Bytes</u> are the <u>two</u>, right-most hex digits. <u>Sectors</u> are represented by the <u>digits left</u> of them.

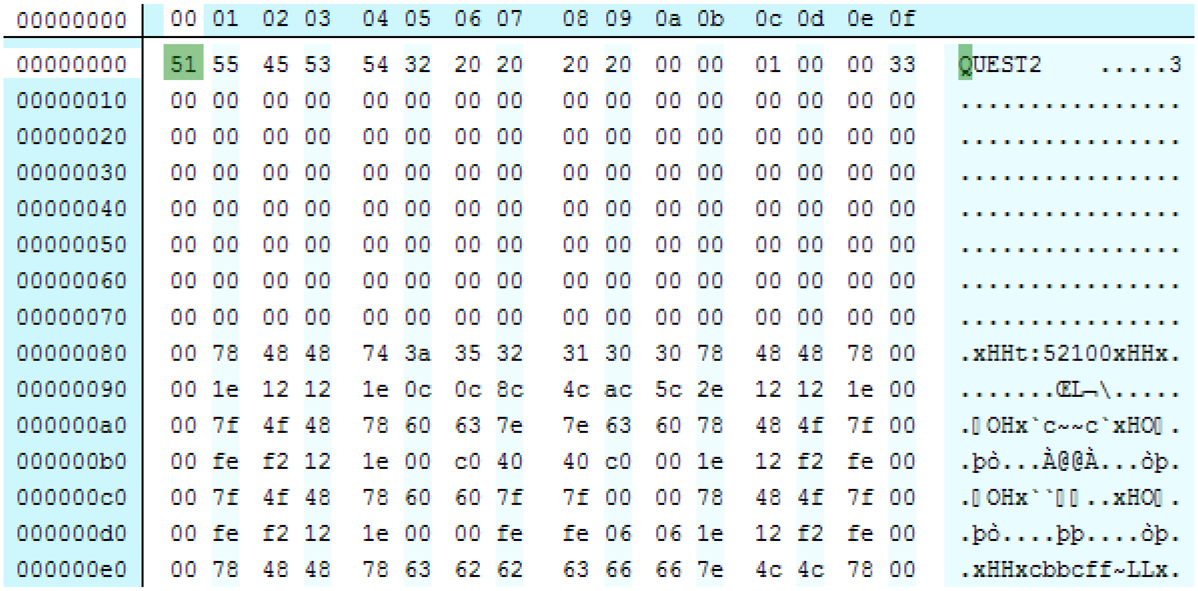
TI diskettes are formatted into 256 byte units (sectors), the bytes ranging in hex from 00 through FF. This sequence begins again with each sector. The number of sectors is determined by the diskette media, the floppy drive capabilities and the disk controller in use. The original TI controller and disk drive combination could work with 90K diskettes, which equal 360 sectors. Correlating references between a Sector/ Byte oriented scheme (of a disk image) to the first through last byte as displayed by a PC Hex Editor is simple - the sector in hex is simply placed before the selected sector byte from 00 to FF!

Thus, Byte 2B in Sector 0030 translates to File Byte 00302B of the disk image as reported by a PC Hex Editor. Byte CF of Sector 0044 translates to File Byte 0044CF of the disk image, etc. The first byte of a TOD database, placed as the first file of a disk image is always on Sector 0022, Byte 00, which a PC Hex Editor would report as Byte 002200.

The third and final approach uses a PC based Hex Editor, which can be used to modify any PC based program. They are commonly referred to as “Hex Editors” instead of Sector Editors as they can typically edit virtually any type of drive as well as computer memory.

For our current purposes these editors will modify both disk images and free standing game databases in V9T9 or TIFILES format located anywhere on your PC drives. A PC Hex Editor commences with the first byte of the disk image or database file. PC based Disk Images are quick, simple to work with and can reduce development time. Based upon recommendations gathered online I experimented with one called ***Hex Editor NEO*** and wish to report “ease of use” and “excellent results”. The necessary database reference guide will require some modification when using a PC based Hex Editor. Another very easy to use PC based editor is ***HXD Hex Editor.***

If you decide to use an individual database file in V9T9 or TIFILES format instead of a disk image with a PC Hex Editor - the byte sequence changes! Unfortunately due to the 128 byte file header information added to the beginning of the file we cannot simply start at byte 00 for editing, but must begin at hex 80, which is hexadecimal for 128. I have included this sequence as an additional reference as space allowed.

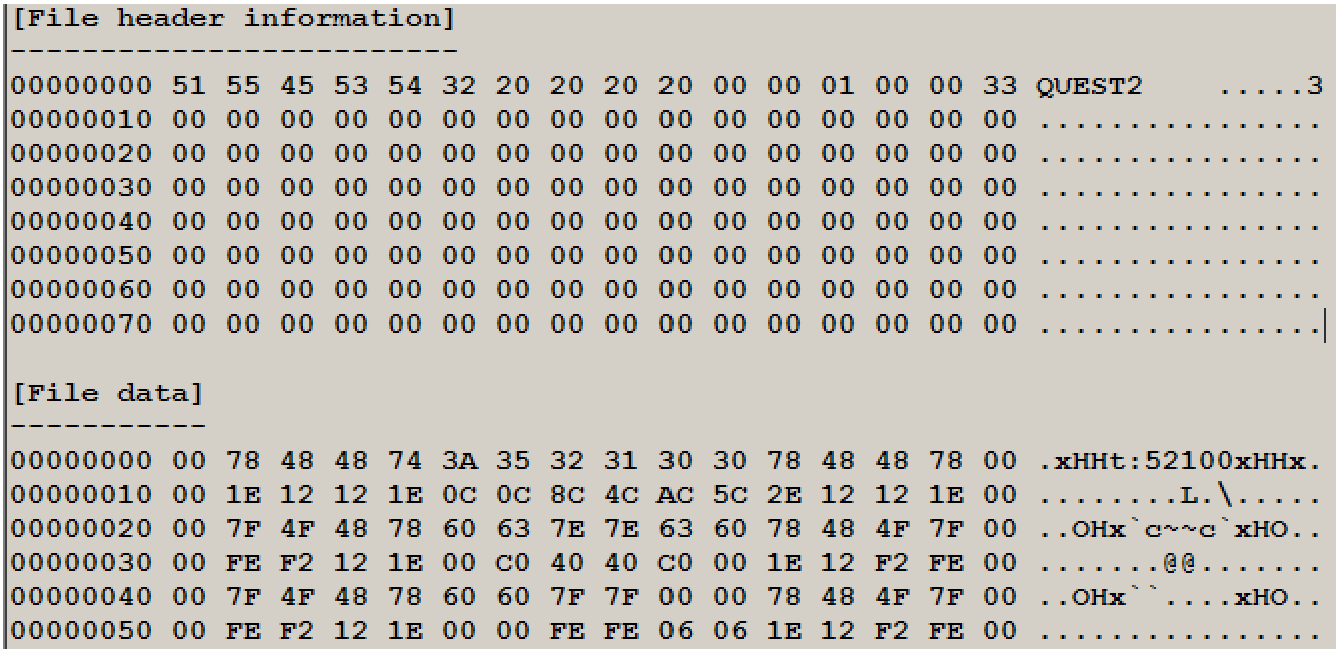


```
00000000 | 00 01 02 03  04 05 06 07  08 09 0a 0b  0c 0d 0e 0f

00000000 | 51 55 45 53  54 32 20 20  20 20 00 00  01 00 00 33   QUEST2    .....3
00000010 | 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000020 | 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000030 | 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000040 | 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000050 | 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000060 | 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000070 | 00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00   ................
00000080 | 00 78 48 48  74 3a 35 32  31 30 30 78  48 48 78 00   .xHHt:52100xHHx.
00000090 | 00 1e 12 12  1e 0c 0c 8c  4c ac 5c 2e  12 12 1e 00   .......ŒL¬\.....
000000a0 | 00 7f 4f 48  78 60 63 7e  7e 63 60 78  48 4f 7f 00   .▯OHx`c~~c`xHO▯.
000000b0 | 00 fe f2 12  1e 00 c0 40  40 c0 00 1e  12 f2 fe 00   .þò...À@@À...òþ.
000000c0 | 00 7f 4f 48  78 60 60 7f  7f 00 00 78  48 4f 7f 00   .▯OHx``▯▯..xHO▯.
000000d0 | 00 fe f2 12  1e 00 00 fe  fe 06 06 1e  12 f2 fe 00   .þò....þþ....òþ.
000000e0 | 00 78 48 48  78 63 62 62  63 66 66 7e  4c 4c 78 00   .xHHxcbbcff~LLx.
```

A Quest Database File preceded by 128 Bytes of File Header Information

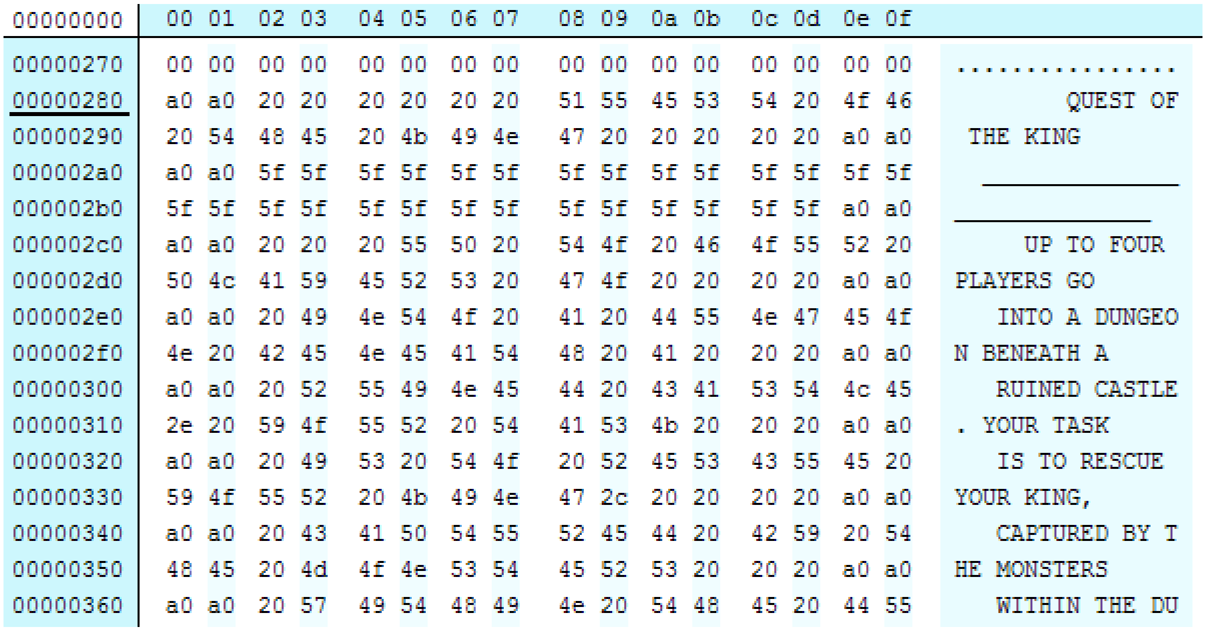
However, this reference scheme is not as complete as its **Sector/ Byte** counterpart due to space and type-width limitations. In addition, the Game Byte # frequently only provides the first byte reference of the string or value to be edited, not the range. The ***Hex/ ASCII Code Listing*** found in the Appendices can be used to determine the correct conversion (in bytes) between TIFILES/ V9T9 formats and a Sector/ Byte format and to plot out the range as necessary.



```
[File header information]
-------------------------
00000000 51 55 45 53 54 32 20 20 20 20 00 00 01 00 00 33 QUEST2    .....3
00000010 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................

[File data]
-----------
00000000 00 78 48 48 74 3A 35 32 31 30 30 78 48 48 78 00 .xHHt:52100xHHx.
00000010 00 1E 12 12 1E 0C 0C 8C 4C AC 5C 2E 12 12 1E 00 ........L.\.....
00000020 00 7F 4F 48 78 60 63 7E 7E 63 60 78 48 4F 7F 00 ..OHx`c~~c`xHO..
00000030 00 FE F2 12 1E 00 C0 40 40 C0 00 1E 12 F2 FE 00 .......@@.......
00000040 00 7F 4F 48 78 60 60 7F 7F 00 00 78 48 4F 7F 00 ..OHx``....xHO..
00000050 00 FE F2 12 1E 00 00 FE FE 06 06 1E 12 F2 FE 00 ................
```

TIDir breakdown of the same



```
00000000 | 00 01  02 03  04 05  06 07   08 09  0a 0b  0c 0d  0e 0f
00000270 | 00 00  00 00  00 00  00 00   00 00  00 00  00 00  00 00   ................
00000280 | a0 a0  20 20  20 20  20 20   51 55  45 53  54 20  4f 46           QUEST OF
00000290 | 20 54  48 45  20 4b  49 4e   47 20  20 20  20 20  a0 a0    THE KING
000002a0 | a0 a0  5f 5f  5f 5f  5f 5f   5f 5f  5f 5f  5f 5f  5f 5f     ______________
000002b0 | 5f 5f  5f 5f  5f 5f  5f 5f   5f 5f  5f 5f  5f 5f  a0 a0   ______________
000002c0 | a0 a0  20 20  20 55  50 20   54 4f  20 46  4f 55  52 20        UP TO FOUR
000002d0 | 50 4c  41 59  45 52  53 20   47 4f  20 20  20 20  a0 a0   PLAYERS GO
000002e0 | a0 a0  20 49  4e 54  4f 20   41 20  44 55  4e 47  45 4f      INTO A DUNGEO
000002f0 | 4e 20  42 45  4e 45  41 54   48 20  41 20  20 20  a0 a0   N BENEATH A
00000300 | a0 a0  20 52  55 49  4e 45   44 20  43 41  53 54  4c 45      RUINED CASTLE
00000310 | 2e 20  59 4f  55 52  20 54   41 53  4b 20  20 20  a0 a0   . YOUR TASK
00000320 | a0 a0  20 49  53 20  54 4f   20 52  45 53  43 55  45 20      IS TO RESCUE
00000330 | 59 4f  55 52  20 4b  49 4e   47 2c  20 20  20 20  a0 a0   YOUR KING,
00000340 | a0 a0  20 43  41 50  54 55   52 45  44 20  42 59  20 54      CAPTURED BY T
00000350 | 48 45  20 4d  4f 4e  53 54   45 52  53 20  20 20  a0 a0   HE MONSTERS
00000360 | a0 a0  20 57  49 54  48 49   4e 20  54 48  45 20  44 55      WITHIN THE DU
```

To summarize: TI based Sector Editor software will report locations on both physical and disk image disks by Sector and Byte. This holds true whether you are using an actual TI99/4a or a PC based emulator. PC software Hex Editors always report from the first Byte of any file: be it a V9T9, TIFILE or a disk image. When using a Hex Editor <u>and</u> disk image files, the last two digits reported represent a specific Byte in hex, from 00 - FF, and the proceeding hex digits represent the Sector. For Hex Editors working on V9T9 or TIFILES, use the File or <u>Game Byte #</u> value to determine the correct editing locations. If this is not specific enough for your needs, use the ***Hex/ ASCII Code Listing*** found in the Appendices.

The following is an <u>approximate</u> breakdown of a Tunnels of Doom database by sectors to provide an indication of how the VDP memory is utilized:

General Database Breakdown	Primary Sector Locations		Total
Game Title & Description:	24-25		2
Graphics (21 Sectors):	22-23	(N,S,E,W symbols; ranged & magical attack graphics)	2
	26	(Characters from saved game)	1
	27-28	ASCII Character definitions	2
	29	Map Graphics	1
	2A-2D	Hallway Graphic space)	4
	2F-30	(Party A/ D) 1 Sector),	1
	37-3A	(Monster A/ D)	4
	4C-50	Rooms & Room Contents graphics	4
	53-54	Current Map	2
Lists & Game Settings: (17 Sectors)	Scattered		17
Monster Data	30-35		5
Specific "Saved" Game Information	2E-2F, 3B-3F		<u>6</u>
Total Game Information (Current & Saved)			51 **Sectors**
Total Game Bytes = Hex 32FF = 13056 Bytes			

Concluding Remarks

As noted, an attempt was made to reference all database locations by an absolute **Game Byte** # and by **Sector/ Byte #**. However, some conversion from **Sector/ Byte** references to **Game Byte #** will likely prove necessary if the latter approach is used. Sector Bytes are in **bold** font to aid in their recognition.

Numbers in parentheses, e.g. (03) that follow Sector/ Byte locations, indicates their value (hence, if unmodified, the default) found in Quest. This is provided as reference to assist designers with establishing baseline values. Currently, the most flexible and comprehensive means of creating a new game database is with a Sector Editor.

When I first undertook serious study of TOD game databases during the early 90's I noted a repeating programming convention, which expressed certain hex numbers in an apparently coded or backwards arrangement. With this arrangement hex: FF=01, FE=02, FD=03, FC=04 and so on. Having neither the benefit of a formal course of study in programming, nor suitable experience, these reversed hex digits presented a mystery. It was plain that they were <u>always</u> used to identify specific intents or options on behalf of the program developer and for lack of a better term I used Reverse Notation in my notes for their designation. This term, or something similar, was necessary to differentiate these hex values from ordinary hex digits as they permitted special, unique and otherwise unavailable options within a game database. It was a full two decades later and thanks entirely to the development of the internet that I learned that these were in fact a form of hex notation commonly used to express negative numbers.

However, Reverse Notation is an easier term to use and somewhat less intimidating than "expressed as a negative number" and so it remained in use throughout my notes, most of which date back to the early 90's. With that said let the reader beware or at the very least be aware!

The original intent was to include design Forms to assist a developer in creating and keeping track of their database. These were primarily intended for use with the various Lists, Graphics and Monster stats.

In essence they would have been the same as those used in the text as examples from ***Quest***, but blank so they could be filled in by the designer. Their inclusion as another Appendix, however, would have had an unwieldy effect on the booklet and so they were omitted. The intent is to post them separately from this work at a later date.

It is certaint that errors will be found within this little booklet. Some of these will be typos; others will result from an incomplete understanding of how the various aspects of a TOD game database work, interact with other settings and from simple short-sightedness on my part. A good example of the latter is the scribble nature of notes taken over the years.. In addition to foreseen shortcomings, there are a number of hex settings that remain unexplained and undefined. There are also a significant number of other memory locations, frequently populated with hex "00" that may provide significant game options in the future. More experimentation and testing is indicated.

In closing one may enquire, "Why was this Guide written at this date and time?" The only reasonable explanation is to preserve this knowledge for current and future TI Home Computer end users… reinventing the wheel is seldom much fun!

Acknowledgements

Victor Steerup - of the Chicago TI User Group (CTIUG). For his tacit reminder last Fall that I had previously affirmed that I would compose this little volume for posterity. (I never knew it would take this long!)

Diane V. - for her patience (?), fine proofing skills and for repeatedly asking, "What are you trying to say here?"

John Behnke - for providing the initial inspiration and the means for new TOD databases.

John Birdwell - for his excellent DSKU Editor.

Mike Brent (tursi) - (of harmlesslion.com fame) for his extremely useful **Classic99**. An even casual perusal of this booklet will reveal how indebted it is to Classic99's video saving features. In particular, the Hallway graphics methodology is extremely difficult to adequately express in writing. The various screenshots say more than 1000 words each.

Fred Kaal - for his excellent, extremely useful and user friedly ***TIDir***. It is an outstanding viewer for every format of TI files. Someday, hopefully, it will also function as an editor.

Chris Schneider (shift838) - for use of his reconstructed VDP memory Pattern Descriptor Table and kindly agreeing to provide a peer review of this work.

Kevin Kenny - last, but certainly not least, for his intial creation of Tunnels of Doom!

I. Graphics in the TOD Environment
An Overview

- The graphic capabilities of ***Tunnels of Doom*** are central to the game's appeal and continued popularity. The program’s ability to design and generate colorful, multi-level, 3-D dungeons along with a wide array of characters and paraphernalia, help to engage player interest. To achieve this over 40% of each game database is devoted to graphic representations.

Anyone who has experience in programming graphics with TI Basic will be pleased to learn that their implementation in the TOD context is quite similar. In TI Basic, the **CALL CHAR** subprogram defines a graphic in the following manner:

CALL CHAR(*character-code, pattern-identifier*)

The *character-code* is an ASCII code from 32-159 and the *pattern-identifier* is a 16-character string expression written in hexadecimal notation. Through this means TI Basic provides the programmer with 127 unique *character-codes* for graphic purposes. By comparison the TOD module can potentially access approximately ***432 character-codes*** for graphic use. How is this possible?

Through an ingenious system of graphic bank switching the module quickly selects from amongst multiple banks of *character-codes* as necessary. To simplify matters all *character-codes* are pre-designated from 00-FF, each of which is allotted space in VDP memory for a 16 character, hexadecimal code, *pattern-identifier*. The locations in VDP memory, where the *pattern-identifiers* are stored/ potentially stored (if data is actually entered), is reflected in their sector/ byte storage on diskette. Each game database is the result of writing the contents of VDP memory to disk. The banks can be tabulated as follows:

Common Bank: Party characters, a set of characters for ASCII codes 32-95 and redefined ASCII sets for characters 96-127 for use as map graphics. (This block includes *character-codes* 00-7F and is common to all three >Hex 7F Graphic Banks).

Three Banks of graphic codes in the >7F range:

Bank #1: All hallway graphics, fountains or statues, doorways, etc. as viewed in the 3-D dungeon exploring hallway mode. (*character-codes* 80-EF).

Bank #2: Rooms, their contents, Quest objects, Monsters, etc. In short, any time the game’s view changes to an overhead view. This includes "Hallway" Monster combat. (*character-codes* 80-EF.)

Bank #3: Graphics for the 4 compass directions, party location indicator, ranged weapon cursor and sequences for ranged and magical weapon attacks. (*character-codes* 80-BF)[1]

and in addition:

Temp: Temporary Dynamic Graphic Workspace. (c*haracter-codes* F0-FB, whose color table bytes are shared in common by both Banks #1 and #2)

Finally, *character-codes* FC-FF, which are unique in that they respond to requests from both Banks #1 and #2.

[1] I have not been able to locate an indication within the game database of how this Bank is referenced. This may actually be Bank #0 or something else entirely.

I. Graphics in the TOD Environment
An Overview

The memory locations for *character-codes* F0-FB are of special interest in that they are used for temporary work space to page in Monster Attack/ Defense graphics, fountains, stairs, vaults and the like. In essence any room item that the player cannot "pick up". *Character-codes* C8-D7 of Bank #1 provides the large graphic blocks necessary to generate the enlarged image counterparts of items seen as you enter a room: monsters, stairs, vaults, etc.

A good portion of Bank #1, which is primarily used for hallway graphics, is unused. It appears that although there was space in VDP RAM to provide for more than one hallway graphic scheme, limitation on module memory made this impractical.

Returning to our analogy of the **CALL CHAR** subprogram of TI Basic, we have yet to identify how the TOD equivalent of the **"CALL CHAR"** command is made. For the Common, Hallway, and part of Bank #3 Graphic Banks the program obtains the necessary hex string from predefined VDP memory locations. Change the hexadecimal data in these locations and the graphic changes (e.g. via a sector editor).

An excellent visual presentation of how VDP memory is used by TOD was recently posted on the web. I am grateful to Chris Schneider for his clear recreation of a Form, which illustrates a concept presented in TI's ***Editor/Assembler Manual*** called a **Pattern Descriptor Table**. Even a casual perusal of this Form (reproduced in **Appendix I**) reveals its relation to the database portion of ***Tunnels of Doom***. What has previously been described as the **Common Graphics Bank** and **Bank #1** (Hallway graphics) can be plotted directly onto this Form as it exists within this Table. What we appreciate as *character-codes* in TI Basic and Extended Basic is here represented by the >hex #, i.e. >00 through >FF memory locations. Thank you shift838!

Color Tables

An understanding of the TI Basic **CALL COLOR** subprogram, where both the FG/ BG colors are specified, assists in understanding their TOD implementation. The sole practical difference is that the colors are designated by hex digits 0-F (transparent to white) instead of the ASCII scheme 1 (transparent) to 16 (white) used in Basic.

Information on graphics color is stored in two discreet tables. One byte of color information represented by two hex digits, set the FG/ BG colors of 8 consecutive *character-codes*. The first Table, which is for graphics Bank #1 in the 80-D7 *character-code* range, permits a Hallway color scheme capable of changing every two levels. This table is located in Sector 004C, bytes 3E to 6F.

The second Table sets the colors for *character-codes* 00-7F of the Common Bank, *codes* 80-D7 of Bank #2 and *codes* D8-FF for both Banks #1& #2. It is located in Sector 004C, bytes 70-8F and is detailed in Chapter IV.

A Reminder! The numbers in **bold**, which precede some of the descriptions are for reference and are used in the hallway examples that follow the **Sector/ Bytes** listings.

II. Graphics in the TOD Environment
Graphics Bank #1 - Hallway Schemes

Color Table #1: for Hallway Characters (Bytes Set FG/ BG Colors)

Hallway Color Table - by Floor #: (Bytes express FG/BG Colors – 1st Hex digit = FG color & 2nd Hex digit = BG color)

Game Byte #	Sector 004C Bytes						
Floors =	1&2	3&4	5&6	7&8	9&10	Char-Codes	
00002ABE	3E	48	52	5C	66	80-87	R & L walls, but not directly ahead or wall/ ceiling interface. Floor.
00002ABF	3F	49	53	5D	67	88-8F	Ceiling. R & L wall/ ceiling interfaces.
00002AC0	40	4A	54	5E	68	90-97	Wall directly ahead. 1st Door/ Wall view directly ahead (4-paces).
00002AC1	41	4B	55	5F	69	98-9F	Door/ Wall interface, side views.
00002AC2	42	4C	56	60	6A	A0-A7	2nd, 3rd & 4th Door direct ahead & side views. Door knob.
00002AC3	43	4D	57	61	6B	A8-AF	Top border of Wall directly ahead 2-paces.
00002AC4	44	4E	58	62	6C	B0-B7	Hallway Fountains
00002AC5	45	4F	59	63	6D	B8-BF	Hallway Fountains
00002AC6	46	50	5A	64	6E	C0-C7	Floor/ door interface to the sides & 1 pace away head-on.
00002AC7	47	51	5B	65	6F	C8-D7	Inside of Room, Steps, Vault as seen when door 1st opens (from outside)

Note: Color Codes for Char-Codes D8 - FF in the Hallway graphics bank are shared with and set by their counterparts in the Rooms and Contents graphics bank.

Hallway Graphics – Graphics Bank #1 (char-Code >7F)

Game Byte #	Sector 002A Bytes	Floor & Walls Colors set in Sector 4C, Bytes: 3E, 48, 52, 5C & 66 (Each byte sets color for 2 Floors)	Char-Codes
00000880	00-07	Walls to your R & L, but not directly in front or above doors	80
00000888	08-0F	Floor of hallways, but not Floor/ Wall interface	81
00000890	10-17	L Floor/ Wall interface	82
00000898	18-1F	R Floor/ Wall interface	83
000008A0	20-27	Blank	84
000008A8	28-2F	Blank	85
000008B0	30-37	Blank	86
000008B8	38-3F	Blank	87
Ceiling & Interface		Colors set in Sector 4C, Bytes: 3F, 49, 53, 5D & 67 (Each byte sets color for 2 Floors)	
000008C0	40-47	Ceiling, but not Ceiling/ Wall interface	88
000008C8	48-4F	1 - L Ceiling/ Wall interface, proximal (more wall than ceiling)	89
000008D0	50-57	2 - L Ceiling/ Wall interface, distal (more ceiling than wall)	8A
000008D8	58-5F	1 - R Ceiling/ Wall interface, proximal (more wall than ceiling)	8B
000008E0	60-67	2 - R Ceiling/ Wall interface, distal (more ceiling than wall)	8C
000008E8	68-6F	Blank	8D
000008F0	70-77	Blank	8E
000008F8	78-7F	Blank	8F
Wall & 1st Door	**View**	Colors set in Sector 4C, Bytes: 40, 4A, 54, 5E & 68 (Each byte sets color for 2 Floors)	
00000900	80-87	Wall viewed directly in front (not to either side)	90
00000908	88-8F	8 - Door/ Wall (Left half - direct ahead) 4 paces - or can be just a large Door \1st Door View	91
00000910	90-97	9 - Door/ Wall (Right half - direct ahead) 4 paces - or can be just a large Door /	92
00000918	98-9F	Blank	93
00000920	A0-A7	Blank	94
00000928	A8-AF	Blank	95
00000930	B0-B7	Blank	96
00000938	B8-BF	Blank	97
Door/ Wall Interface to R & L		Colors set in Sector 4C, Bytes: 41, 4B, 55, 5F & 69 (Each byte sets color for 2 Floors)	
00000940	C0-C7	1 - L corner, above Door \ (always most proximal (more door than wall)	98
00000948	C8-CF	2 - Center block, above Door \ **door is viewed on your L**	99
00000950	D0-D7	3 - Center block, above Door /	9A
00000958	D8-DF	4 - R corner, above Door / (always most distal (more wall than door)	9B
00000960	E0-E7	1 - R corner, above Door \ (always most proximal (more door than wall)	9C
00000968	E8-EF	2 - Center block, above Door \ **door is viewed on your R**	9D
00000970	F0-F7	3 - Center block, above Door /	9E
00000978	F8-FF	4 - L corner, above Door / (always most distal (more wall than door)	9F

II. Graphics in the TOD Environment
Graphics Bank #1 - Hallway Schemes

Note: Walls and the Ceiling & Floor, Doors and the Wall above & Floor have interfaces. These are tapered in a proximal to distal orientation, which helps to create the 3-D hallway perspective used in TOD. The numbers in bold preceding descriptions represent this proximal to distal order placement. See examples.

Hallway Graphics (Especially Door and Hall Fountains)

Game	Sector 002B		
Byte #	Bytes	Colors set in Sector 4C, Bytes: 42, 4C, 56, 60 & 6A (Each byte sets color for 2 Floors)	Char-Codes
00000980	00-07	**7 -** Bottom ½ of Door both to R & L - 1st view; most of Door: 2nd, 3rd & 4th view, both ahead, R & L	A0
00000988	08-0F	Blank	A1
00000990	10-17	Blank	A2
00000998	18-1F	Doorknob, 2 & 1-pace away	A3
000009A0	20-27	**5 -** Top ½ of a Door on your L, 1st view; R top corner of Door on your L - 3rd & 4th view	A4
000009A8	28-2F	**6 -** Top ½ of a Door on your R, 1st view; L top corner of Door on your R - 3rd & 4th view	A5
000009B0	30-37	Top of Door, viewed directly ahead, 1-pace away	A6
000009B8	38-3F	Bottom of Door, viewed directly ahead, 1-pace away	A7
		Colors set in Sector 4C, Bytes: 43, 4D, 57, 61 & 6B (Each byte sets color for 2 Floors)	
000009C0	40-47	Top border of Wall directly ahead, viewed from 2-paces	A8
000009C8	48-4F	Blank	A9
000009D0	50-57	Blank	AA
000009D8	58-5F	Blank	AB
000009E0	60-67	Blank	AC
000009E8	68-6F	Blank	AD
000009F0	70-77	Blank	AE
000009F8	78-7F	Blank	AF
		Colors set in Sector 4C, Bytes: 44, 4E, 58, 62 & 6C (Each byte sets color for 2 Floors)	
00000A00	80-87	Center of Top of Fountain, viewed 3, 2 & 1-pace away	B0
00000A08	88-8F	Center row of Fountain Top, Design, (4 of them), 1-pace away	B1
00000A10	90-97	L half of Fountain 4-paces away, L Top of Fountain 3-paces, L Top corner 2 & 1-pace	B2
00000A18	98-9F	R half of Fountain 4-paces away, R Top of Fountain 3-paces, R Top corner 2 & 1-pace	B3
00000A20	A0-A7	Blank	B4
00000A28	A8-AF	Blank	B5
00000A30	B0-B7	Blank	B6
00000A38	B8-BF	Blank	B7
		Colors set in Sector 4C, Bytes: 45, 4F, 59, 63 & 6D (Each byte sets color for 2 Floors)	
00000A40	C0-C7	Base of Fountain - 3, 2 & 1 pace away (mixed with floor background 2 & 1-pace away)	B8
00000A48	C8-CF	L bottom corner, of Top of Fountain, 2 & 1-pace away	B9
00000A50	D0-D7	R bottom corner, of Top of Fountain, 2 & 1-pace away	BA
00000A58	D8-DF	Top row of Center of Fountain, 1-pace away (Quest = solid bar)	BB
00000A60	E0-E7	Blank	BC
00000A68	E8-EF	Blank	BD
00000A70	F0-F7	Blank	BE
00000A78	F8-FF	Blank	BF

Game	Sector 002C Door interface		
Byte #	Bytes	Colors set in Sector 4C, Bytes: 46, 50, 5A, 64 & 6E (Each byte sets color for 2 Floors)	Char-Codes
00000A80	00-07	Door/ Floor interface, Doors on L side of Hall	C0
00000A88	08-0F	Door/ Floor interface, Doors on R side of Hall	C1
00000A90	10-17	Blank	C2
00000A98	18-1F	Blank	C3
00000AA0	20-27	Blank	C4
00000AA8	28-2F	Blank	C5
00000AB0	30-37	Blank	C6
00000AB8	38-3F	Blank	C7

II. Graphics in the TOD Environment
Graphics Bank #1 - Hallway Schemes

Large Block Graphics used when entering Rooms or when Hallway Monsters Appear

Game	Sector 002C		
Byte #	Bytes	Colors set in Sector 4C, Bytes: 47, 51, 5B, 65 & 6F (Each byte sets color for 2 Floors)	Char-Codes
00000AC0	40-47	(space char, (20) used when you enter an empty room)	C8
00000AC8	48-4F	0F0F 0F0F 0000 0000	C9
00000AD0	50-57	F0F0 F0F0 0000 0000	CA
00000AD8	58-5F	FFFF FFFF 0000 0000	CB
00000AE0	60-67	0000 0000 0F0F 0F0F	CC
00000AE8	68-6F	0F0F 0F0F 0F0F 0F0F	CD
00000AF0	70-77	F0F0 F0F0 0F0F 0F0F	CE
00000AF8	78-7F	FFFF FFFF 0F0F 0F0F	CF
00000B00	80-87	0000 0000 F0F0 F0F0	D0
00000B08	88-8F	0F0F 0F0F F0F0 F0F0	D1
00000B10	90-97	F0F0 F0F0 F0F0 F0F0	D2
00000B18	98-9F	FFFF FFFF F0F0 F0F0	D3
00000B20	A0-A7	0000 0000 FFFF FFFF	D4
00000B28	A8-AF	0F0F 0F0F FFFF FFFF	D5
00000B30	B0-B7	F0F0 F0F0 FFFF FFFF	D6
00000B38	B8-BF	FFFF FFFF FFFF FFFF	D7

Note: These large character blocks are used to create the (large) facsimile of the monster(s) seen in the room, as viewed when you first open the door. If no monsters are present, then a large graphic of an item in the room, e.g. a Vault or Stairway, is shown. If the room is completely empty then it is filled with the Space **Char-Code** – Hex 20 by **Char-Code** (C8).

Definable Hallway Graphic Space

Game	Sector 002C		
Byte #	Bytes	Color set in Sector 4C, Byte 8B (Color Byte is shared)	Char-Codes
00000B40	C0-C7	Blank	D8
00000B48	C8-CF	Blank	D9
00000B50	D0-D7	Blank	DA
00000B58	D8-DF	Blank	DB
00000B60	E0-E7	Blank	DC
00000B68	E8-EF	Blank	DD
00000B70	F0-F7	Blank	DE
00000B78	F8-FF	Blank	DF

Definable Hallway Graphic Space

Game	Sector 002D		
Byte #	Bytes	Color set in Sector 4C, Byte 8C (Color Byte is shared)	Char-Codes
00000B80	00-07	Blank	E0
00000B88	08-0F	Blank	E1
00000B90	10-17	Blank	E2
00000B98	18-1F	Blank	E3
00000BA0	20-27	Blank	E4
00000BA8	28-2F	Blank	E5
00000BB0	30-37	Blank	E6
00000BB8	38-3F	Blank	E7
		Color set in Sector 4C, Byte 8D (Color Byte is shared)	
00000BC0	40-47	Blank	E8
00000BC8	48-4F	Blank	E9
00000BD0	50-57	Blank	EA
0000OBD8	58-5F	Blank	EB
0000OBE0	60-67	Blank	EC
00000BE8	68-6F	Blank	ED
00000BF0	70-77	Blank	EE
00000BF8	78-7F	Blank	EF

Note: The Color Bytes assigned in Sector 004C: 8B, 8C and 8D, control the FG/ BG coloring of Graphic Banks 1 & 2 (Hallways and Room Contents) for their respective Char-Codes D8-EF.

II. Graphics in the TOD Environment
Graphics Bank #1 - Hallway Schemes

Dynamic Game Graphic Workspace, Non-Definable (Sector 002D, Bytes 80-DF)

Game Byte #	Sector 002D Bytes	Color set by Sector 4C, Byte 8E (Color Byte is shared, but over written)	Char-Codes
00000C00	80-87	Reserved	F0
00000C08	88-8F	Reserved	F1
00000C10	90-97	Reserved	F2
00000C18	98-9F	Reserved	F3
00000C20	A0-A7	Reserved	F4
00000C28	A8-AF	Reserved	F5
00000C30	B0-B7	Reserved	F6
00000C38	B8-BF	Reserved	F7
		Color set by Sector 4C, Byte 8F (Color Byte is shared, but over written)	
00000C40	C0-C7	Reserved	F8
00000C48	C8-CF	Reserved	F9
00000C50	D0-D7	Reserved	FA
00000C58	D8-DF	Reserved	FB

Note:

80-9F	Monster Defense Graphic paged in, Fountain & Living Statue codes paged in when viewing "Hallway Fountains". If a color besides '1E' is used, it will only persist until a key is pressed before reset by the module back to '1E' (Black on Red).	(F0-F3)
A0-BF	Duplicate Monster Defense Graphic, Stairs Up/ Down, Vault - all paged into workspace.	(F4-F7)
C0-DF	Monster Attack Graphic - paged in	(F8-FB)

Definable Game Graphic Workspace

Game Byte #	Sector 002D Bytes	Color set by Sector 4C, Byte 8F (Color Byte is shared)	Char-Codes
00000C60	E0-E7		FC
00000C68	E8-EF		FD
00000C70	F0-F7		FE
00000C78	F8-FF		FF

Note: Bytes E0-FF Definable for use, however, if Color Code is changed from '6E', the new color will flicker briefly the first time the Monster Attack Graphic comes into play in each room. FC-FF

Notes on Bytes:

It is best not to use Bytes 80 - FF to avoid potential conflicts and undesired effects. If **Char-Codes** FC-FF are required for graphic purposes in a game, retain the Black on Red, Hex '1E', color scheme.

Char-Codes F0 through FF are shared in common by both Hallway and Room Contents Graphic Banks (1 & 2). This includes both memory workspace and color control Bytes.

Sector 002D: In both 'Quest for the King' and 'Pennies', this sector contains volumes of Hex Digits. They differ from one another in Hex values, but are alike in that the values ***never*** change. These ***appear*** to be artifacts from the game 'Creation' .

Total Program Bytes to current Section: **BFF** = 3,071 Bytes.

Color Codes **Note:** (The two hex digits of each byte control the Foreground/ Background colors as per TI Basic and XB. Call Color subprogram)

0 Transparent	4 Dark Blue	8 Medium Red	C Dark Green
1 Black	5 Light Blue	9 Light Red	D Magenta
2 Medium Green	6 Dark Red	A Dark Yellow	E Gray
3 Light Green	7 Cyan	B Light Yellow	F White

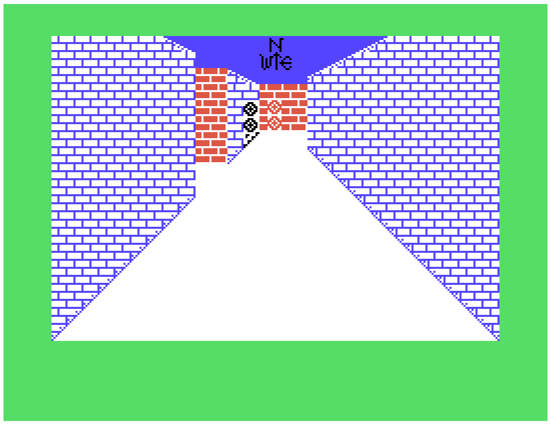
The main Dungeon components of a ***Tunnels of Doom*** database are: Walls, Ceiling, Floor, Doors and Fountains. All of these features, Fountains excepted, have an "optional" tapering interface that is used to generate the 3-D distance perspective seen in Hallway views. Due to their dynamic nature some of these interfaces are difficult to tabulate or describe. I have taken the liberty to present both actual Hallway views and the same view with selected graphics replaced by "numbers" - hereafter referred to as "labels", as an aid in comprehension.

III. Graphics in the TOD Environment
Sample Hallways - ~~Paint~~ Graphics by Number

Hallway Example from Halls - Door View #1

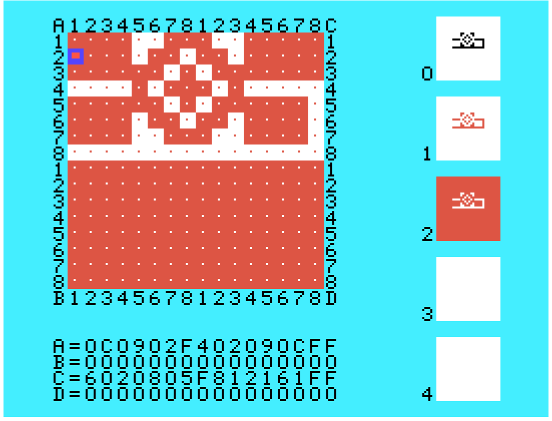
Game	Sector 002A		
Byte #	Bytes	Colors set in Sector 4C, Bytes: 40, 4A, 54, 5E & 68 (Each byte sets color for 2 Floors)	Char-Codes
00000900	80-87	wall viewed directly in front (not to either side)	90
00000908	88-8F	Door/ wall (Left half - direct ahead) 4 paces - or can be just a large Door \1st Door View	91
00000910	90-97	Door/ wall (Right half - direct ahead) 4 paces - or can be just a large Door /	92

Doors – 4 Paces



First Hallway Door View

L-Half of Door R-Half of Door (White on Dark Red)



```
A=0C0902F402090CFF
B=0000000000000000
C=6020805F812161FF
D=0000000000000000
```

Two character-codes of first view

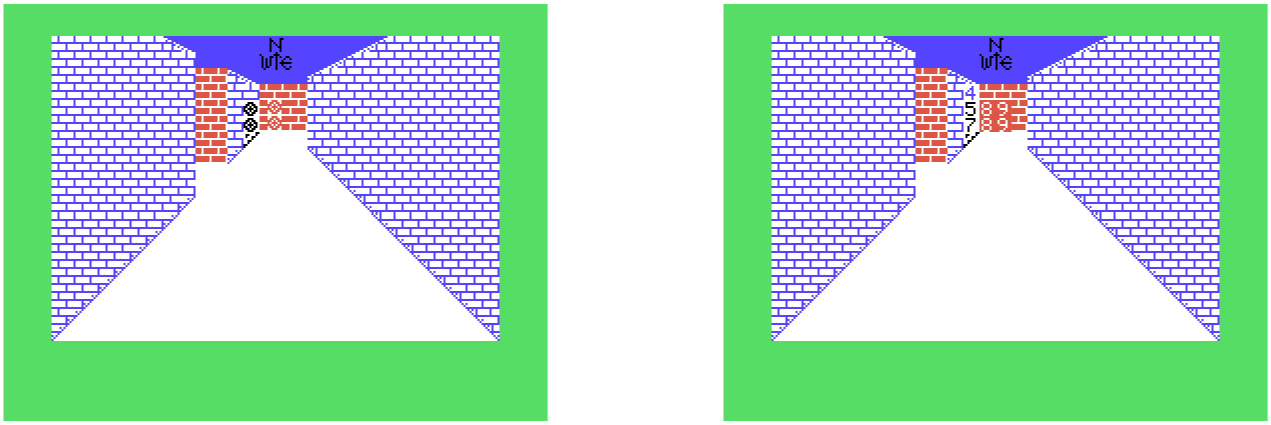
```
Label   Game        Sector 002A Bytes - Hex display of corresponding Bytes from Halls
        Byte #          00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
        00000880    00  10 10 10 FF 01 01 01 FF 00 00 00 00 00 00 00 00  Floor
        00000890    10  01 0A 04 F8 10 A0 40 80 A8 48 28 1F 08 05 02 01
        000008A0    20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        000008B0    30  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        000008C0    40  FF FF FF FF FF FF FF FF FF 3F 4F FB 0C 09 08 FF  Ceiling
        000008D0    50  FF FF FF FF FF 3F 4F 93 FF FC F2 DF 21 A1 21 FF  Ceiling/ Wall Interface
        000008E0    60  FF FF FF FF FF FC F2 C9 00 00 00 00 00 00 00 00  First Door View #1
        000008F0    70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    (Directly Ahead)
#8      00000900    80  10 10 10 FF 01 01 01 FF 0C 09 02 F4 02 09 0C FF  Bottom L ½ of Door
#9      00000910    90  60 20 80 5F 81 21 61 FF 00 00 00 00 00 00 00 00  Bottom R ½ of Door
        00000920    A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        00000930    B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
        00000940    C0  10 10 10 FF 01 01 01 FF 10 10 10 FF 01 01 01 FF
        00000950    D0  10 10 10 FF 01 01 01 FF 10 10 10 FF 01 01 01 FF
        00000960    E0  10 10 10 FF 01 01 01 FF 10 10 10 FF 01 01 01 FF
        00000970    F0  10 10 10 FF 01 01 01 FF 10 10 10 FF 01 01 01 FF
```

In this example of the 1st Door view, seen directly ahead, both the left and right door halves and the adjoining walls are defined. The color is set to F6 (in **Sector 004C**, byte **40**), a white foreground on a dark red background, which colors the door design and the adjoining brick wall. The TOD program automatically doubles this design, increasing the door's height as seen. (Screenshots from ***Halls***.)

The door on the left is colored 1F (**Sector 004C**, byte **42**), black on a white background.

Game	Sector 002B		
Byte #	Bytes	Colors set in Sector 4C, Bytes: 42, 4C, 56, 60 & 6A (Each byte sets color for 2 Floors)	Char-Codes
00000980	00-07	Bottom ½ of Door both to R & L - 1st view; most of Door: 2nd, 3rd & 4th view, both ahead & R & L	A0

Sample Hallways - ~~Paint~~ Graphics by Number



First Door View #2 **(L & R Sides of Hall)**

```
Label Game          Sector 002B - Hex display of corresponding Bytes from Halls
      Byte #        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
#7    00000980 00   00 38 6C D6 BA D6 6C 38 00 00 00 00 00 00 00 00  7=Primary Door Design
      00000990 10   00 00 00 00 00 00 00 00 00 18 3C 5A 81 81 42 3C  Doorknob (3rd & 4th Door view)
5&6   000009A0 20   00 38 6C D6 BA D6 6C 38 00 38 6C D6 BA D6 6C 38  5=Top ½ Door on your L
      000009B0 30   FF FF 00 FF FF 00 FF FF FF FF 00 FF FF 00 FF FF  6=Top ½ Door on your R
      000009C0 40   FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00
      000009D0 50   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      000009E0 60   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      000009F0 70   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00000A00 80   7E FF FF FF FF FF FF FF FF BD 5A E7 66 E7 66 C3
      00000A10 90   00 4C C2 A9 A3 3F 7F 7F 00 32 47 95 C5 FC FE FE
      00000A20 A0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00000A30 B0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00000A40 C0   FF FF E3 C9 DD C9 E3 FF 80 80 C0 C0 A0 B0 FC BF
      00000A50 D0   01 01 03 03 05 0D 3F FD FF 66 24 00 00 00 00 00
      00000A60 E0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      00000A70 F0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
                                                                     Char-Codes
      000009A0 20-27  Top ½ of Door on your L, 1st view; R top corner of Door on your L - 3rd & 4th view   A4
      000009A8 28-2F  Top ½ of Door on your R, 1st view; L top corner of Door on your R - 3rd & 4th view   A5
```

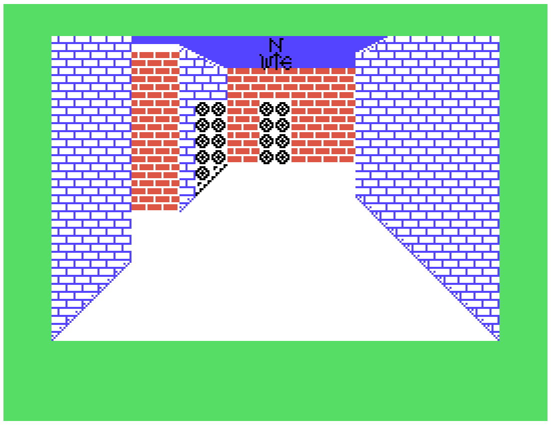
While the entire 1st Door view seen directly ahead is defined in **Sector 002A**, its side view counterparts are defined in **Sector 002B**. The bytes represented by label #7 represent the bottom half of the door when viewed on either side of the hall. They are also used as the primary Door design in all subsequent views. The top half of a side viewed Door (1st view) is defined by the bytes represented by #5 for doors on the left side of the hall and by #6 for the right side. Both the hex definitions represented by the numbers 5 and 6 are reused as a corner of the Door design in the 3rd and 4th Door views. The #4 above the Door represents the Door/ Wall Interface.

Door View #2

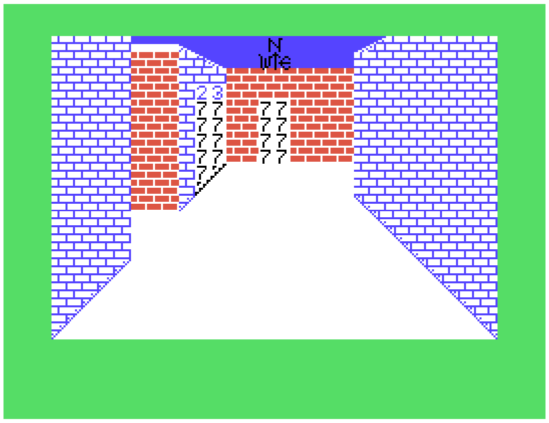
The 2nd hallway Door views (3-paces from door) utilize a single character-code for implementation. These hex bytes are also stored in **Sector 002B**, label #7 above. The two Doors as a series of #7's on the next page, illustrates how this works. The designs of both the head on and side views of the doors are represented here.

```
Game           Sector 002B
Byte #         Bytes   Colors set in Sector 4C, Bytes:  42, 4C, 56, 60 & 6A (Each byte sets color for 2 Floors)   Char-Codes
00000980       00-07   2nd Door view. All of Door 3-paces away, direct and side.                                  A0
                       (Used as primary design in all 4 Door views.)
00000980       00-07   00 38 6C D6 BA D6 6C 38 00 00 00 00 00 00 00 00 (All 4 door views.)                       A0
```

III. Graphics in the TOD Environment
Sample Hallways - ~~Paint~~ <u>Graphics</u> by Number



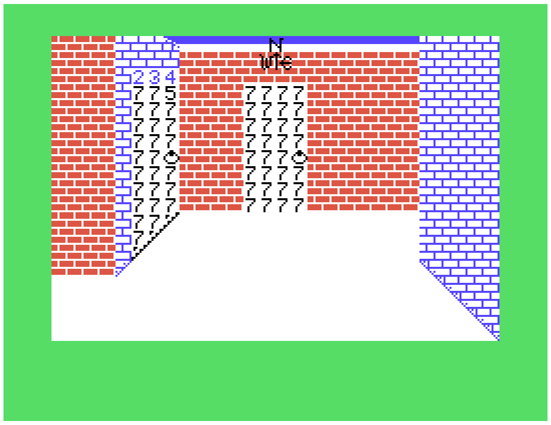
Second Hallway Door View



Used for both side & direct views

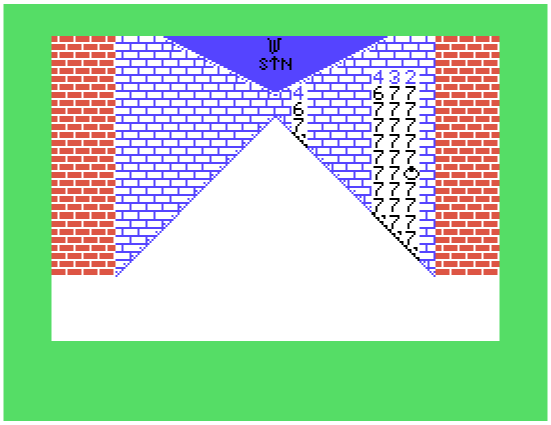
Here we see the 2nd hallway Door view (3 paces away), defined by a single *character-code*, **A0** (label #7 above - see **Sector 002B**). The door colors are now the same and the adjoining wall segments are no longer included. The single design is tiled for the entire Door in both views. Note how the side view door perspective is created by adding an additional image of the design in the column proximal to the viewer. For this reason, a vertical design, or a discrete stand alone graphic works best. Horizontal lines and patterns have a skewed appearance when viewed to the side. Interfaces will be discussed later (#2 & #3).

<u>*Door View #3*</u>



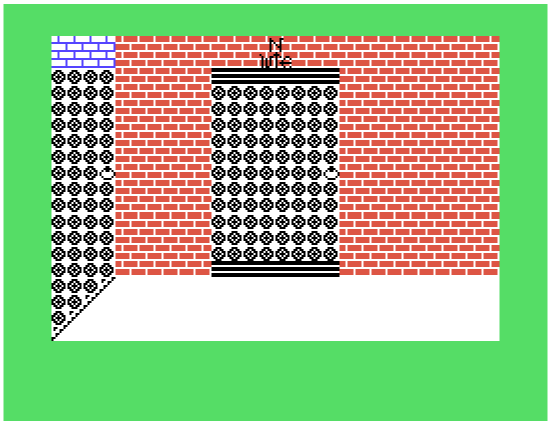
The same *character-code*, **A0** (label **#7)** is again used in the 3rd hallway Door view. The tiling is controlled by the TOD program. Aside from an *additional* Door column, two new factors are introduced. The first is that a <u>doorknob</u> is added by replacing one of the door pattern characters. These are always on the right side of a door. The second is the reuse of *character-codes* **A4** (label #5) and **A5** (label #6) on the Top of the distal columns as a design option. Note once more how the side viewed door image uses a descending order of 10 proximal, 9 medial and 8 distal characters to create the hallway distance perspective. The numbers 2, 3 and 4 above the Door represents the Interface.

Third Hallway Door Views



This is an additional Door view, this time of right hand hallway Doors. The furthest Door is Door view #1 (4-paces away) and the closest is Door view #3 (2-paces). Note the progression of label <u>#6</u> from the entire top ½ of the Door in view #1, to the top upper distal column graphic in view #3. The graphic represented by labels #5 and #6 are always placed at the top of the distal column of a Door. #5 for left sided hallway doors and #6 for right sided hallway doors.

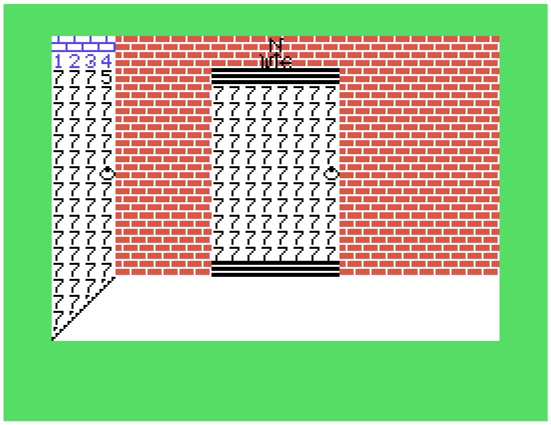
Door View #4

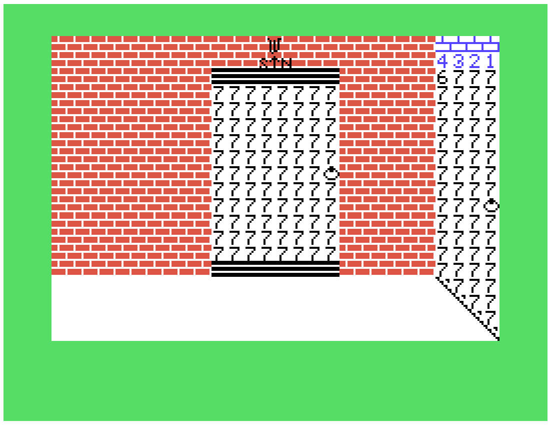


The 4th and final hallway Door view uses the same *character-code*, AO. It doubles the basic Door pattern horizontally, increases the door height by 3 characters and adds a door border design above and below (this is optional) in the direct ahead view. The side view utilizes the same proximal to distal descending number of graphics to create a distance perspective.

Note again the retention of *character-codes* **A4** and **A5**, labels #5 and #6 respectively. An optional door border may be added to the final Door view on doors seen directly ahead.

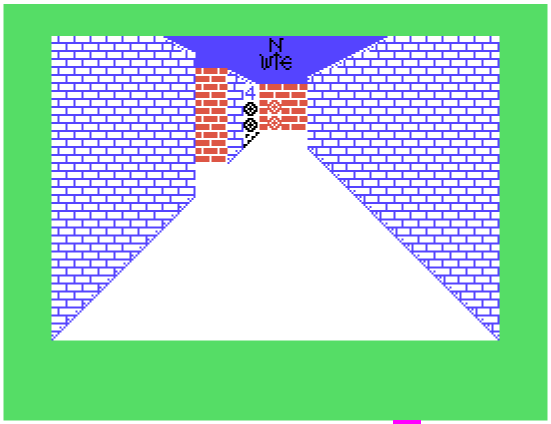
Fourth Hallway Door Views



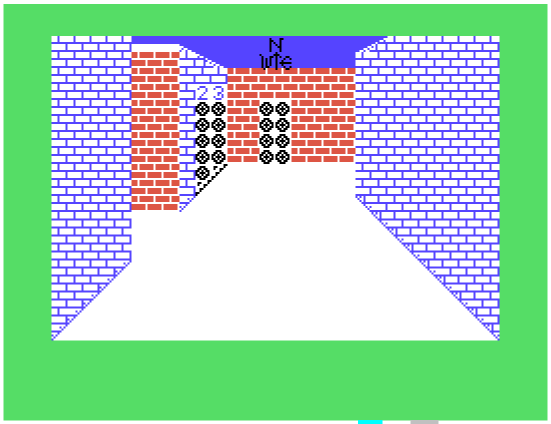


Hallway Examples from Halls - Door & Wall Interface (Left side)

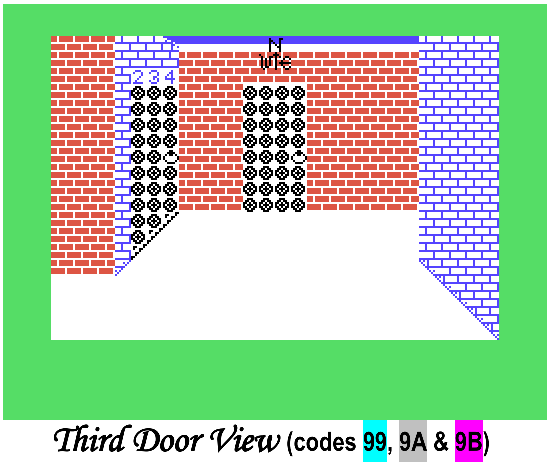
Game Byte #	Sector 002A Bytes		Char-Codes
Side view Door/ Wall Interface		Colors set in Sector 4C, Bytes: 41, 4B, 55, 5F & 69 (Each byte sets color for 2 Floors)	
00000940	C0-C7	1 - L corner, above Door \ always most proximal (more door than wall)	98
00000948	C8-CF	2 - Center block, above Door \	99
00000950	D0-D7	3 - Center block, above Door /door is viewed on your L	9A
00000958	D8-DF	4 - R corner, above Door / always most distal (more wall than door)	9B
00000960	E0-E7	1 - R corner, above Door \ always most proximal (more door than wall)	9C
00000968	E8-EF	2 - Center block, above Door \	9D
00000970	F0-F7	3 - Center block, above Door /door is viewed on your R	9E
00000978	F8-FF	4 - L corner, above Door / always most distal (more wall than door)	9F



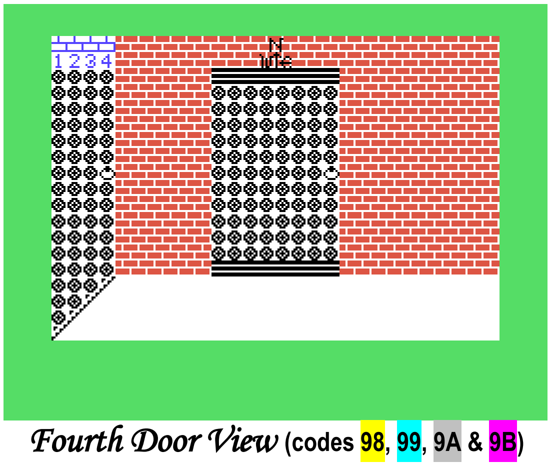
First Door View (code 9B)



Second Door View (codes 99 & 9A)

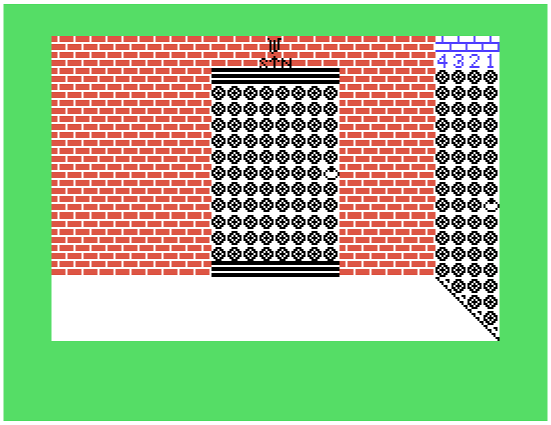


Third Door View (codes 99, 9A & 9B)



Fourth Door View (codes 98, 99, 9A & 9B)

The sequence above shows the same progression of Door views, but with the focus on the Door/ Wall (side view) Interface. ***Halls*** did not utilize this particular interface as it did not work out very well with a brick hallway design! Horizontally oriented designs can be a bit tough to implement. The brick hex codes that were originally placed in these interface bytes are now replaced with the *pattern-identifiers* of the numbers 1- 4 (labels #1-4) to visually demonstrate how TOD manipulates this interface. The number 1 is always the most proximally placed (when used) and represents the least Wall portion of the interface viewed and is also the least tapered. The number 4 represents the opposite - most: tapered, distal and Wall portion. The graphics represented by the numbers are always placed in an ascending order.

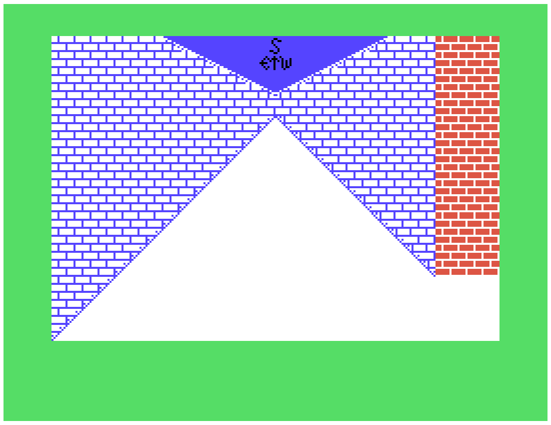


Fourth Door View after Turning Left 90⁰

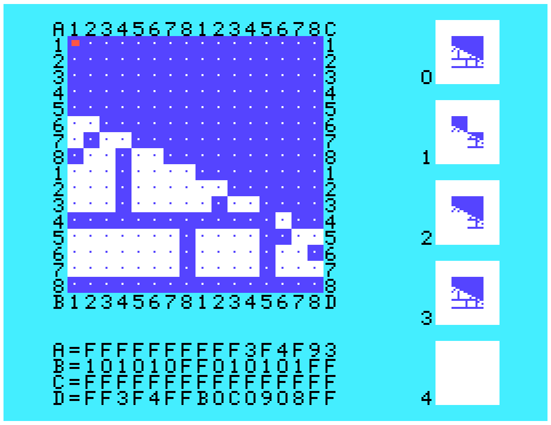
This final interface view shows the same two doors after turning left 90⁰. The interface to your right follows the exact sequence as already noted.

Hallway Examples from Halls - Ceiling & Wall Interface (Left side)

Game Byte #	Sector 002A Bytes	Colors set in Sector 4C, Bytes: 3F, 49, 53, 5D & 67 (Each byte sets color for 2 Floors)	
000008C0	40-47	Ceiling, but not along wall edges	88
000008C8	48-4F	L Ceiling/ wall interface, proximal – more wall than ceiling	89
000008D0	50-57	L Ceiling/ wall interface, distal – more ceiling than wall	8A



Example of Ceiling/ Wall Interface



```
A=FFFFFFFFFF3F4F93
B=101010FF010101FF
C=FFFFFFFFFFFFFFFF
D=FF3F4FFB0C0908FF
```

Progression of Wall/ Ceiling Interface

The examples above demonstrate the basic progression of the wall/ ceiling interface for the left-hand side of the hallway. The small graphic #1 is the proximal (upper left) and distal (lower right) sections of the interface, *character-codes* 89 and 8A. Graphic #2 adds the section of ceiling, *character-code* 88, which joins to this part of the interface (upper right). #3 adds the wall character, *character-code* 80 that joins with the interface (lower left). As the wall is generated by a single *character-code*, you will need to adapt the right and left interfaces as necessary.

The Ceiling/ Wall interface is as good time as any to discuss color choices for your Dungeon. What follows will also pertain to all interfaces. Frequently, the design of an interface is radically different "top and bottom" as shown in small graphic #1 (above right). Similarly, the Door bottom/ Floor designs will also be rather different. While this is relatively simple to do as in the example above, coloring an interface can be tricky as the entire interface uses the same FG/ BG color. In the example on the left we see a blue Ceiling, a blue and white brick Wall and a white Floor. For these interfaces to match they must use blue, white or a combination of both.

III. Graphics in the TOD Environment
Sample Hallways - ~~Paint~~ Graphics by Number

Quest offers a simple, but effective solution by making the BG color of one character the FG color of the other. This is simple to do when the walls are plain and without design. In the example just seen the entire Ceiling is set to FFFFFFFF (all foreground) making it easy to match the blue (hex 4F) brick design of the walls. Conversely, the white background of the walls (hex 4F) matches the white background of the Floor, set to 00000000 (all background). (Set in Sector 004C, bytes 3E and 3F).

If a unique interface is used, which is not intended to blend two components such as Ceiling/ Floor then a more varied color scheme is possible within a game.

Hallway Examples from Halls - Wall

Game	Sector 002A		
Byte #	Bytes	Colors set in Sector 4C, Bytes: 3E, 48, 52, 5C & 66 (Each byte sets color for 2 Floors)	Char-Codes
00000880	00-07	Walls to your R & L, but not directly in front or above doors	80

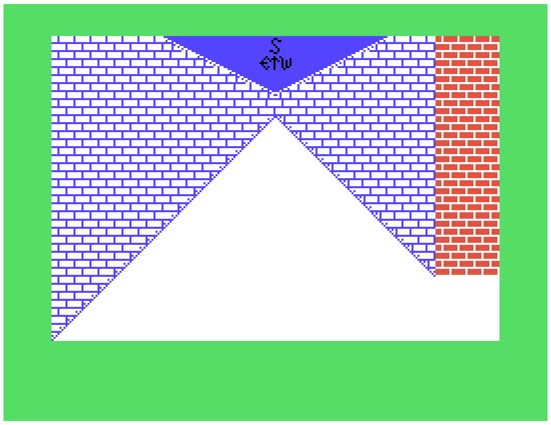
00000880 10 10 10 FF 01 01 01 FF 00 00 00 00 00 00 00 00 (*pattern-identifier* of *character-code* 80 - hallway wall.)

Remember! These instructions for dungeon design are not nearly as complicated as they may seem, especially for the interfaces. The game designer need only insert the correct *pattern-identifier* in the appropriate memory (**Sector**) location. The TOD program manages all of the graphic character placements. This includes graphics tiling, transpositions and adjusting sets for head-on and side hallway views.

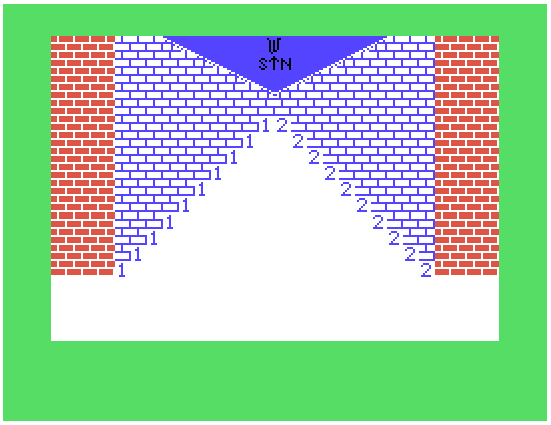
The lower screen interfaces; for Wall/ Floor and Door/ Floor are simpler designs having only one *character-code* each, for each side of the Hallway. See all example below.

Hallway Examples from Quest - Floor/ wall Interface

Game	Sector 002A	Floor & Walls	
Byte #	Bytes	Colors set in Sector 4C, Bytes: 3E, 48, 52, 5C & 66 (Each byte sets color for 2 Floors)	Char-Codes
00000880	00-07	Walls to your R & L, but not directly in front or above doors	80
00000888	08-0F	Floor of hallways, but not Floor/ Wall interface	81
00000890	10-17	1 - L Floor/ Wall interface	82
00000898	18-1F	2 - R Floor/ Wall interface	83



Two views of a Floor/ Wall Interface.



A Floor/ Wall Interface with graphics

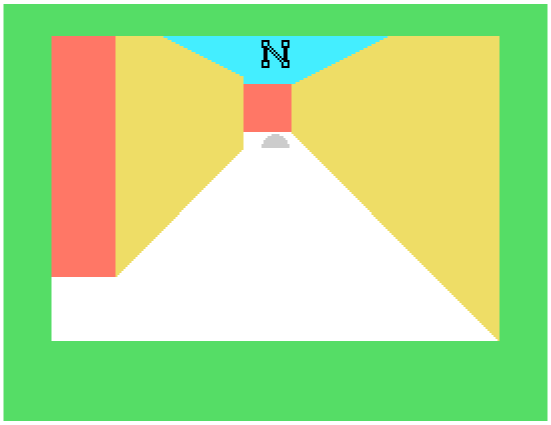
A Similar Interface represented by labels

Our final example of Hallway Graphics will demonstrate the progression of Fountain (top section) displays. The following uses the designs from ***Quest*** as illustration.

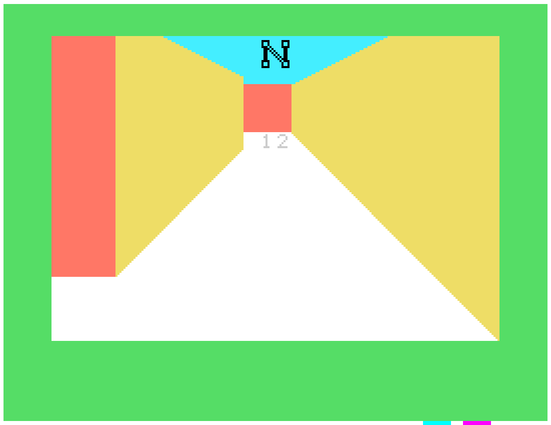
Hallway Examples from Quest - Fountain Tops

Game Byte #	Sector 002B Bytes	Colors set in Sector 4C, Bytes: 44, 4E, 58, 62 & 6C (Each byte sets color for 2 Floors)	
00000A00	80-87	3 – Center of Top of Fountain, viewed 3, 2 & 1-pace away	B0
00000A08	88-8F	Center row of Fountain Top, Design, (4 of them), 1-pace away	B1
00000A10	90-97	1 – L half of Fountain 4-paces away, L Top of Fountain 3-paces, L Top corner 2 & 1-pace	B2
00000A18	98-9F	2 – R half of Fountain 4-paces away, R Top of Fountain 3-paces, R Top corner 2 & 1-pace	B3
00000A20	A0-A7	Blank	B4
00000A28	A8-AF	Blank	B5
00000A30	B0-B7	Blank	B6
00000A38	B8-BF	Blank	B7
Primarily Fountain Base		Colors set in Sector 4C, Bytes: 45, 4F, 59, 63 & 6D (Each byte sets color for 2 Floors)	
00000A40	C0-C7	Base of Fountain - 3, 2 & 1 pace away (mixed with floor background 2 & 1-pace away)	B8
00000A48	C8-CF	L bottom corner, of Top of Fountain, 2 & 1-pace away	B9
00000A50	D0-D7	R bottom corner, of Top of Fountain, 2 & 1-pace away	BA
00000A58	D8-DF	Top row of Center of Fountain, 1-pace away (Quest = solid bar)	BB
00000A60	E0-E7	Blank	BC
00000A68	E8-EF	Blank	BD
00000A70	F0-F7	Blank	BE
00000A78	F8-FF	Blank	BF

```
Game        Sector 002B
Byte #        00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F        B0(#3)
00000A00   8  FF FF FF FF FF FF FF FF FF FF BB 55 EF EF EF FF    B1=top design x4
00000A10   9  00 03 0F 1F 3F 3F 7F 7F 00 C0 F0 F8 FC FC FE FE    B2(#1) B3(#2)
00000A20   A  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000A30   B  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000A40   C  FF FF FF FF FF FF FF FF 80 80 C0 C0 E0 F0 FC FF
00000A50   D  01 01 03 03 07 0F 3F FF FF FF 00 00 00 00 00 00    BB
00000A60   E  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00000A70   F  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```



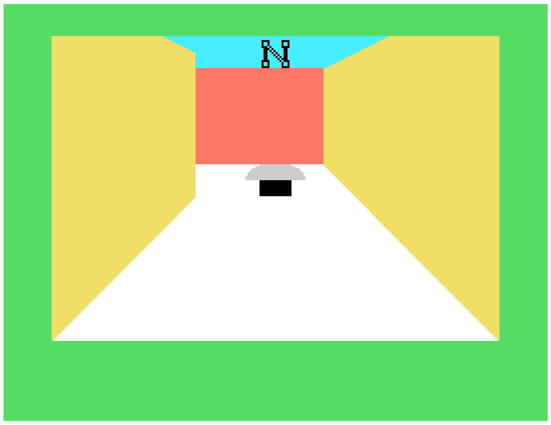
First Fountain View



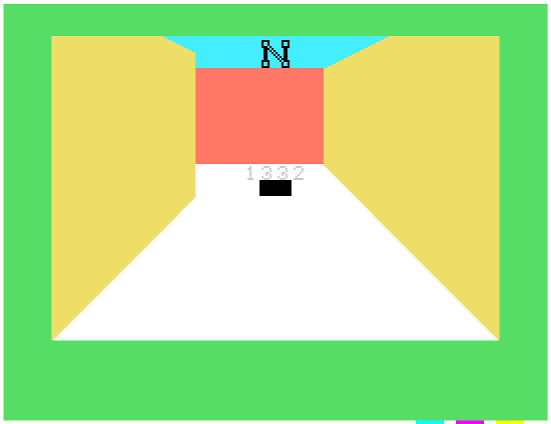
First Fountain View (codes B2 B3)

The TOD construction of hallway Fountains or Statues is very similar to that of hallway Doors. A basic design or shape is presented at a distance forming a core portion. The core design is then added to and elaborated upon in subsequent views.

Sample Hallways - ~~Paint~~ Graphics by Number

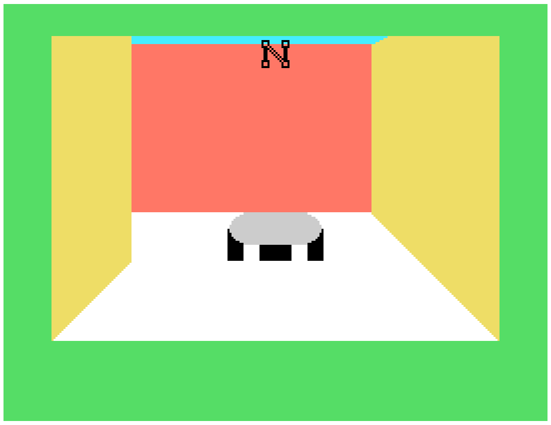


Second Fountain View

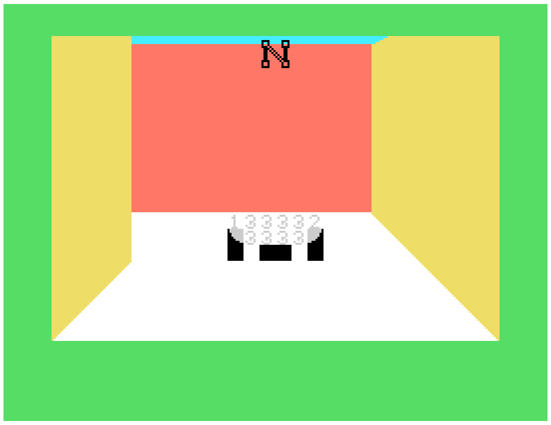


Second Fountain View (codes **B2** **B3** **B0**)

The top is formed with the addition of **code B0** (doubled). The base of the Fountain is added - **code B8**.

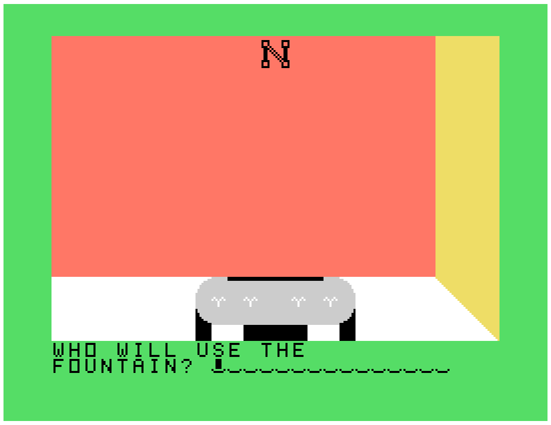


Third Fountain View



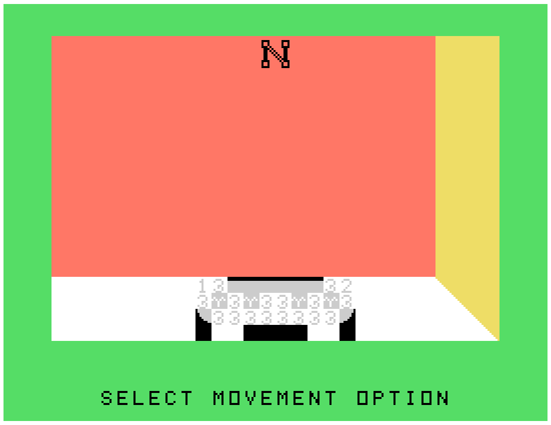
Third Fountain View (note use of label #3)

Label #3, code **B0** is proliferated (quadrupled). The base is elaborated - **codes B9** & **BA** to L & R bases.



```
WHO WILL USE THE
FOUNTAIN? ▮_______________
```

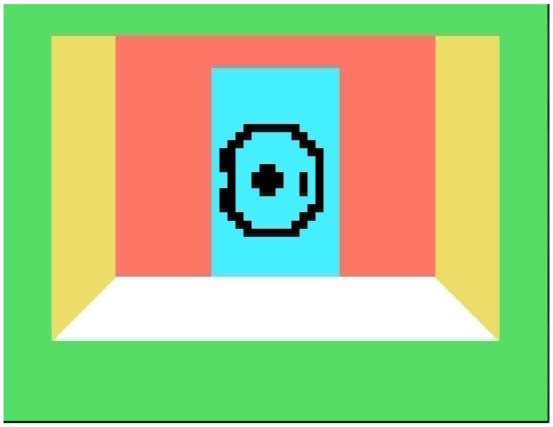
Fourth Fountain View



```
SELECT MOVEMENT OPTION
```

Final Fountain View – as labels
(With Top Design code B1 (x4) & Top Row code BB)

III. Graphics in the TOD Environment
Sample Hallways - ~~Paint~~ Graphics by Number



A large portion of Sector 002C consists of a group of graphic blocks used to generate the enlarged images of room items; stairs, vaults, monsters and the like. These are briefly seen when the room door first opens, but before you enter the room - assuming any items were present. If no items are present then *character-code* `C8`, the space character, fills the room entrance view.

Example from Quest

Large Block Graphics - 16 Character Sets
used on entering a Room or when a Hallway Monster Appears

Game	Sector 002C		
Byte #	**Bytes**	Colors set in Sector 4C, Bytes: 47, 51, 5B, 65 & 6F (Each byte sets color for 2 Floors)	**Char-Codes**
00000AC0	**40-47**	(space char, (20) used when you enter an empty room)	C8
00000AC8	**48-4F**	0F0F 0F0F 0000 0000	C9
00000AD0	**50-57**	F0F0 F0F0 0000 0000	CA
00000AD8	**58-5F**	FFFF FFFF 0000 0000	CB
00000AE0	**60-67**	0000 0000 0F0F 0F0F	CC
00000AE8	**68-6F**	0F0F 0F0F 0F0F 0F0F	CD
00000AF0	**70-77**	F0F0 F0F0 0F0F 0F0F	CE
00000AF8	**78-7F**	FFFF FFFF 0F0F 0F0F	CF
00000B00	**80-87**	0000 0000 F0F0 F0F0	D0
00000B08	**88-8F**	0F0F 0F0F F0F0 F0F0	D1
00000B10	**90-97**	F0F0 F0F0 F0F0 F0F0	D2
00000B18	**98-9F**	FFFF FFFF F0F0 F0F0	D3
00000B20	**A0-A7**	0000 0000 FFFF FFFF	D4
00000B28	**A8-AF**	0F0F 0F0F FFFF FFFF	D5
00000B30	**B0-B7**	F0F0 F0F0 FFFF FFFF	D6
00000B38	**B8-BF**	FFFF FFFF FFFF FFFF	D7

IV. Graphics in the TOD Environment
The Common Graphics Bank

The Common Graphics Bank is compatible with all >7F graphic *character-code* banks so that text and ASCII symbols can be mixed with screen graphics. TOD has its own defined set of ASCII characters (capital letters only), which are loaded from the game base to VDP memory on startup. The first 16 bytes of ***Color Table #2***, located on **Sector 004C**, are used to set *Foreground/ Background* colors of the Common Graphic Bank *character-codes*.

Color Table #2: for Common Graphics Bank & Graphics Bank #2 (Bytes Set FG/ BG Colors)

Common Bank - Player Characters Classes 1 - 4, Regular ASCII Characters 32 - 95, Floor Map

Game	Sector 004C	Quest			
Byte #	Bytes	Value	Char-Codes	Description	Sector Found
00002AF0	**70**	4E	00-07	Character Class #1 D/ A	**Sector 0026**
00002AF1	**71**	DE	08-0F	Character Class #2 D/ A	
00002AF2	**72**	CE	10-17	Character Class #3 D/ A	
00002AF3	**73**	DE	18-1F	Character Class #4 D/ A	
00002AF4	**74**	13	20-27	ASCII Codes 32-39	**Sector 0027**
00002AF5	**75**	13	28-2B	ASCII Codes 40-47	
00002AF6	**76**	13	30-37	ASCII Codes 48-55	
00002AF7	**77**	13	38-3F	ASCII Codes 56-63	
00002AF8	**78**	13	40-47	ASCII Codes 64-71	**Sector 0028**
00002AF9	**79**	13	48-4F	ASCII Codes 72-79	
00002AFA	**7A**	13	50-57	ASCII Codes 80-87	
00002AFB	**7B**	13	58-5F	ASCII Codes 88-95	
00002AFC	**7C**	1E	60-67	ASCII Codes 96-103	\ **Sector 0029** Map Graphics of areas explored
00002AFD	**7D**	1E	68-6F	ASCII Codes 104-111	/
00002AFE	**7E**	EE	70-77	ASCII Codes 112-119	\ Map Graphics color before Map is found
00002AFF	**7F**	EE	78-7F	ASCII Codes 120-127	/ (i.e. Gray on Gray)

Graphics Bank #2 – Rooms and all Room Contents

00002B00	**80**	1E	80-87	Bytes: 90-CF	**Sector 004C**
00002B01	**81**	1E	88-8F	Bytes: D0-0F	**Sector 004D** (Byte 81 does not work? See byte 8E)
00002B02	**82**	1E	90-97	Bytes: 10-4F	(Byte 82 works on Ground Floor only - Game Logo)
00002B03	**83**	1E	98-9F	Bytes: 50-8F	
00002B04	**84**	1E	A0-A7	Bytes: 90-CF	
00002B05	**85**	1A	A8-AF	Bytes: D0-0F	**Sector 004E**
00002B06	**86**	17	B0-B7	Bytes: 10-4F	
00002B07	**87**	12	B8-BF	Bytes: 50-8F	
00002B08	**88**	1A	C0-C7	Bytes: 90-CF	
00002B09	**89**	16	C8-CF	Bytes: D0-0F	**Sector 004F**
00002B0A	**8A**	1E	D0-D7	Bytes: 10-4F	
00002B0B	**8B**	1E	D8-DF	Bytes: 50-8F	(shared with Hallway from here to Char-Code 'FF')
00002B0C	**8C**	16	E0-E7	Bytes: 90-CF	
00002B0D	**8D**	00	E8-EF	Bytes: D0-0F	**Sector 0050**
00002B0E	**8E**	1E	F0-F7	Bytes: 80-BF	**Sector 002D** Stairs & Monsters-D (Monsters initially only)
00002B0F	**8F**	6E	F8-FF	Bytes: C0-FF	**F8-FB** Monsters-A **FC-FF** - free space

***Color Codes* Note**: (The two hex digits of each byte control the Foreground/ Background colors as per TI Basic and XB.)

0	Transparent	4	Dark Blue	8	Medium Red	C	Dark Green
1	Black	5	Light Blue	9	Light Red	D	Magenta
2	Medium Green	6	Dark Red	A	Dark Yellow	E	Gray
3	Light Green	7	Cyan	B	Light Yellow	F	White

Saved Character Graphics – Common Graphics Bank (common to all three >7F Banks)

Game	Sector 0026 (Character colors of Current or from last Saved Game)			
Byte #	Bytes		(Sector 004C)	Char-Codes
00000480	**00-1F**	Character #1 Defense pose	Color set by Byte: 70	00-03
000004A0	**20-3F**	Character #1 Attack pose		04-07
000004C0	**40-5F**	Character #2 Defense pose	Color set by Byte: 71	08-0B
000004E0	**60-7F**	Character #2 Attack pose		0C-0F
00000500	**80-9F**	Character #3 Defense pose	Color set by Byte: 72	10-13
00000520	**A0-BF**	Character #3 Attack pose		14-17
00000540	**C0-DF**	Character #4 Defense pose	Color set by Byte: 73	18-1B
00000560	**E0-FF**	Character #4 Attack pose		1C-1F

Note: These Character graphic colors are used when you select "Continue Current Game", or they are offered for use if the number of Players selected is the same as those of the last saved game. Otherwise they are defined anew each new game.

Redefined ASCII Characters 32-63 (As Character-Codes - all Colors set in Sector 004C)

Game	Sector 0027		Color set by Byte: 74	Game	Sector 0027		Color set by Byte: 76
Byte #	Bytes		Char-Codes	Byte #	Bytes		Char-Codes
00000580	**00-07**	(space)	20	00000600	**80-87**	0	30
00000588	**08-0F**	!	21	00000608	**88-8F**	1	31
00000590	**10-17**	"	22	00000610	**90-97**	2	32
00000598	**18-1F**	#	23	00000618	**98-9F**	3	33
000005A0	**20-27**	$	24	00000620	**A0-A7**	4	34
0000o5A8	**28-2F**	%	25	00000628	**A8-AF**	5	35
000005B0	**30-37**	&	26	00000630	**B0-B7**	6	36
000005B8	**38-3F**	'	27	00000638	**B8-BF**	7	37
			Color set by Byte: 75				Color set by Byte: 77
000005C0	**40-47**	(	28	00000640	**C0-C7**	8	38
000005C8	**48-4F**	)	29	00000648	**C8-CF**	9	39
000005D0	**50-57**	*	2A	00000650	**D0-D7**	:	3A
000005D8	**58-5F**	+	2B	00000658	**D8-DF**	;	3B
000005E0	**60-67**	,	2C	00000660	**E0-E7**	↑ (up arrow)	3C
000005E8	**68-6F**	-	2D	00000668	**E8-EF**	=	3D
000005F0	**70-77**	.	2E	00000670	**F0-F7**	↓ (down arrow)	3E
000005F8	**78-7F**	/	2F	00000678	**F8-FF**	?	3F

Redefined ASCII Characters 64-95 (As Character-Codes - all Colors set in Sector 004C)

Game	Sector 0028		Color set by Byte: 78	Game	Sector 0028		Color set by Byte: 7A
Byte #	Bytes		Char-Codes	Byte #	Bytes		Char-Codes
00000680	**00-07**	@	40	00000700	**80-87**	P	50
00000688	**08-0F**	A	41	00000708	**88-8F**	Q	51
00000690	**10-17**	B	42	00000710	**90-97**	R	52
00000698	**18-1F**	C	43	00000718	**98-9F**	S	53
000006A0	**20-27**	D	44	00000720	**A0-A7**	T	54
000006A8	**28-2F**	E	45	00000728	**A8-AF**	U	55
000006B0	**30-37**	F	46	00000730	**B0-B7**	V	56
000006B8	**38-3F**	G	47	00000738	**B8-BF**	W	57
			Color set by Byte: 79				Color set by Byte: 7B
000006C0	**40-47**	H	48	00000740	**C0-C7**	X	58
000006C8	**48-4F**	I	49	00000748	**C8-CF**	Y	59
000006D0	**50-57**	J	4A	00000750	**D0-D7**	Z	5A
000006D8	**58-5F**	K	4B	00000758	**D8-DF**	©	5B
000006E0	**60-67**	L	4C	00000760	**E0-E7**	→ (right arrow)	5C
000006E8	**68-6F**	M	4D	00000768	**E8-EF**	_/ (cursor space)	5D
000006F0	**70-77**	N	4E	00000770	**F0-F7**	\■/ (cursor)	5E
000006F8	**78-7F**	O	4F	00000778	**F8-FF**	▬ (thick bar)	5F

Note: The Char-Codes 20 - 5F are supplied by the module. On game boot-up they are transferred from the module to VDP memory and then saved to these Sectors with a 'Saved Game'. If redefined they will simply revert back to these sets.

Map Symbol Graphics (ASCII Characters 96-127 - all Colors set in Sector 004C)

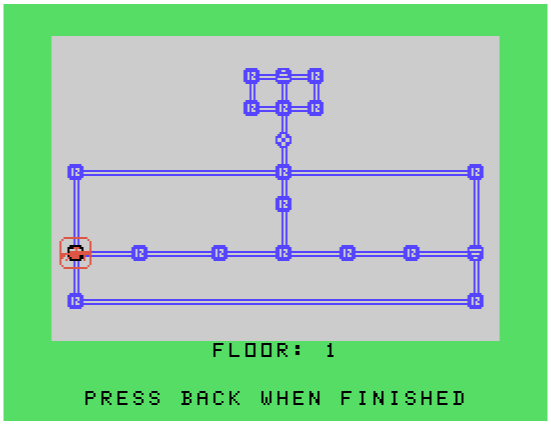
Map Graphics/ Colors - Explored Areas				**Map Graphics/ Colors - Unexplored (Default = Gray on Gray = Invisible)**			
Game Byte #	Sector 0029 Bytes		**Color set by Byte: 7C** Char-Codes	Game Byte #	Sector 0029 Bytes		**Color set by Byte: 7E** Char-Codes
00000780	00-07	═	60	00000800	80-87	═	70
00000788	08-0F	║	61	00000808	88-8F	║	71
00000790	10-17	╬	62	00000810	90-97	╬	72
00000798	18-1F	╩	63	00000818	98-9F	╩	73
000007A0	20-27	╦	64	00000820	A0-A7	╦	74
000007A8	28-2F	╣	65	00000828	A8-AF	╣	75
000007B0	30-37	╠	66	00000830	B0-B7	╠	76
000007B8	38-3F	▣ Room Symbol	67	00000838	B8-BF	▣ Room Symbol	77
			Color set by Byte: 7D				**Color set by Byte: 7F**
000007C0	40-47	⇧ Stairs Up	68	00000840	C0-C7	⇧ Stairs Up	78
000007C8	48-4F	⇩ Stairs Down	69	00000848	C8-CF	⇩ Stairs Down	79
000007D0	50-57	⊕ Hall Fountain	6A	00000850	D0-D7	⊕ Hall Fountain	7A
000007D8	58-5F	Floor & Map Texture	6B	00000858	D8-DF	Floor & Map Texture	7B
000007E0	60-67	→ Right arrow	6C	00000860	E0-E7	→ Right arrow	7C
000007E8	68-6F	_/ Cursor Space	6D	00000868	E8-EF	_/ Cursor Space	7D
000007F0	70-77	Blank	6E	00000870	F0-F7	Blank	7E
000007F8	78-7F	Blank	6F	00000878	F8-FF	Blank	7F

Note: Remember, graphic *character-codes* 00 - 7F are common to all >7F *character-code* banks.

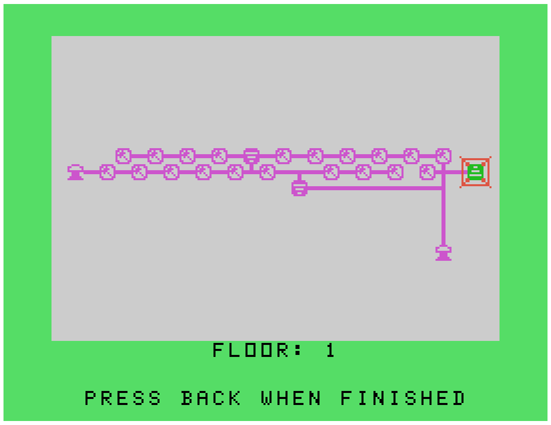
Map Graphic *character-codes* are duplicated so that explored and unexplored areas of a floor can be different colors. ***Quest*** uses **hex EE**, or gray on gray, to define the "colors" of the unexplored areas of a Map. This "gray on gray" coloring causes the unexplored areas to be "invisible" on the Map. When a map is found the colors change to blue on gray, which is handled by the TOD game program.

However, this is not the only possible Map coloring scheme. The coloring of unexplored areas can be altered from "gray on gray" to any combination desired, which would make the floor plan always visible. For example, many commercial buildings, hotels, government facilities and the like have an "Emergency Escape Route" plan prominent on every floor., which could justify this usage in a game base. However, one could still require that a Map (perhaps with an authorization code thereon) is necessary to descend deeper within the Facility.

In any event, once a Map is found the blue on gray coloring, as defined within the TOD program, would be utilized. In addition, there does not appear to be a means of altering the term "Map" from within a game base.



Examples of Map Graphics



V. Graphics in the TOD Environment
Graphics Bank #2 - Room Design & Contents

Color Table #2 allocates an additional 16 bytes of information (**Sector 004C**) to define the *Foreground/ Background* colors of *Graphic Bank #2 character-codes*.

Color Table #2

For Graphics Bank #2 – Rooms and their Contents

Game	Sector 004C	Quest			
Byte #	Bytes	Value	Char-Codes	Description	Sector Found
00002B00	**80**	1E	80-87	Bytes: 90-CF	**Sector 004C**
00002B01	**81**	1E	88-8F	Bytes: D0-0F	**Sector 004D** (Byte 81 does not work? See byte 8E)
00002B02	**82**	1E	90-97	Bytes: 10-4F	(Byte 82 works on Ground Floor only - Game Logo)
00002B03	**83**	1E	98-9F	Bytes: 50-8F	
00002B04	**84**	1E	A0-A7	Bytes: 90-CF	
00002B05	**85**	1A	A8-AF	Bytes: D0-0F	**Sector 004E**
00002B06	**86**	17	B0-B7	Bytes: 10-4F	
00002B07	**87**	12	B8-BF	Bytes: 50-8F	
00002B08	**88**	1A	C0-C7	Bytes: 90-CF	
00002B09	**89**	16	C8-CF	Bytes: D0-0F	**Sector 004F**
00002B0A	**8A**	1E	D0-D7	Bytes: 10-4F	
00002B0B	**8B**	1E	D8-DF	Bytes: 50-8F	(shared with Hallway from here to Char-Code 'FF')
00002B0C	**8C**	16	E0-E7	Bytes: 90-CF	
00002B0D	**8D**	00	E8-EF	Bytes: D0- 0F	**Sector 0050**
00002B0E	**8E**	1E	F0-F7	Bytes: 80-BF	**Sector 002D** Stairs & Monsters-D (Monsters initially only)
00002B0F	**8F**	6E	F8-FF	Bytes: C0-FF	**F8-FB** Monsters-A **FC-FF** - free space

***Color Codes* Note**: (The two hex digits of each byte control the *Foreground/ Background* colors as per TI Basic and XB.)

0	Transparent	4	Dark Blue	8	Medium Red	C	Dark Green
1	Black	5	Light Blue	9	Light Red	D	Magenta
2	Medium Green	6	Dark Red	A	Dark Yellow	E	Gray
3	Light Green	7	Cyan	B	Light Yellow	F	White

Remember, the hex color codes placed into bytes **8A** through **8F**, will alter the coloring schemes of both *Graphic Banks #1 and #2* in the **D8** through **FF** *character-code* range. Check your results!

V. Graphics in the TOD Environment
Graphics Bank #2 - Room Design & Contents

Rooms and their Contents – Graphics Bank #2 (char-Codes 80-FF)

The following graphics do not appear to be re-locatable: ***Stairs, Game Logo, Vault & Room Design Graphics*** (note #1)

Game	**Sector 004C**			
Byte #	**Bytes**	Definable Graphic space, Color set by Byte 80	**Quest** = 1E	**Pennies** = 1E
00002B10	**90-97**	Char-Code 80	\	Star Dust L Bottom
00002B18	**98-9F**	Char-Code 81	\	Stardust R Bottom
00002B20	**A0-A7**	Char-Code 82	/ Room Fountain	Fishbowl L Top
00002B28	**A8-AF**	Char-Code 83	/	Fishbowl R Top
00002B30	**B0-B7**	Char-Code 84	\	Face of Parakeet
00002B28	**B8-BF**	Char-Code 85	\Living Statue	\
00002B40	**C0-C7**	Char-Code 86	/	/ Same as
00002B48	**C8-CF**	Char-Code 87	/	/ Quest
00002B50	**D0-EF**	Char-Codes 88-8B Color set by Byte 8E	Stairs Up (= 1E)	Stairs Up (= 1E)
00002B70	**F0-0F**	Char-Codes 8C-8F **(Sector 004D)**	Stairs Down	Stairs Down
Game	**Sector 004D - *Graphic Character Space for Room Contents***			
Byte #	**Bytes**			
00002B90	**10-2F**	Char-Codes 90-93 (01) Color set by Byte 82	Game Logo	Game Logo
00002BB0	**30-4F**	Char-Codes 94-97 (02)	Vault (note #2)	(#94) Magic Glass L Bottom/ Vault
	Bytes	Definable Graphic space, Color set by Byte 83	**Quest** = 1E	**Pennies** = 17
00002BD0	**50-57**	Char-Code 98\	(98 – 9F Blank)	Picture L Top
00002BD8	**58-5F**	Char-Code 99 \ (03)		Picture R Top
00002BE0	**60-67**	Char-Code 9A /		Fishbowl L Bottom
00002BE8	**68-6F**	Char-Code 9B/		Fishbowl R Bottom
00002BF0	**70-77**	Char-Code 9C\		Magic Glass R Bottom
00002BF8	**78-7F**	Char-Code 9D \ (04) (etc. see note #3)		
00002C00	**80-87**	Char-Code 9E /		
00002C08	**88-8F**	Char-Code 9F/		
	Bytes	Definable Graphic space, Color set by Byte 84	**Quest** = 1E	**Pennies** = 18
00002C10	**90-97**	Char-Code A0	(A0 – A7 Blank)	Top L Top
00002C18	**98-9F**	Char-Code A1		Top R Top
00002C20	**A0-A7**	Char-Code A2		Magic Glass L Top
00002C28	**A8-AF**	Char-Code A3		\
00002C30	**B0-B7**	Char-Code A4		\ Pennies Graphic
00002C38	**B8-BF**	Char-Code A5		/
00002C40	**C0-C7**	Char-Code A6		/
00002C48	**C8-CF**	Char-Code A7		
	Bytes	Definable Graphic space, Color set by Byte 85	**Quest** = 1A	**Pennies** = 12
00002C50	**D0-D7**	Char-Code A8	\	Picture L Bottom
00002C58	**D8-DF**	Char-Code A9	\ Armor	Picture R Bottom
00002C60	**E0-E7**	Char-Code AA	/	Magic Glass L Top
00002C68	**E8-EF**	Char-Code AB	/	Same as Quest
00002C70	**F0-F7**	Char-Code AC	\	\
00002C78	**F8-FF**	Char-Code AD	\ Shield	\ Same as Quest
00002C80	**00-07**	Char-Code AE (Sector 004E)	/	/
00002C88	**08-0F**	Char-Code AF	/	/
	Sector 004E - *Graphic Character Space for Room Contents*			
	Bytes	Definable Graphic space, Color set by Byte 86	**Quest** = 17	**Pennies** = 6E
00002C90	**10-17**	Char-Code B0	\	
00002C98	**18-1F**	Char-Code B1	\ Map	Puppy
00002CA0	**20-27**	Char-Code B2	/	
00002CA8	**28-2F**	Char-Code B3	/	
00002CB0	**30-37**	Char-Code B4	Wand L Side	Puppy L Bottom (same as B3)
00002CB8	**38-3F**	Char-Code B5	Wand R Side	Parakeet L Bottom
00002CC0	**40-47**	Char-Code B6	Potion – Bottom	Parakeet L Bottom
00002CC8	**48-4F**	Char-Code B7	Orb R Bottom	

V. Graphics in the TOD Environment
Graphics Bank #2 - Room Design & Contents

Game	Sector 004E - *Graphic Character Space for Room Contents*				
Byte #	Bytes	Definable Graphic space, Color set by	Byte 87	**Quest** = 12	**Pennies** = 5E
00002CD0	50-57	Char-Code B8		Scroll L Side	\
00002CD8	58-5F	Char-Code B9		Scroll R Side	\ Snail
00002CE0	60-67	Char-Code BA		Chest L Bottom	/
00002CE8	68-6F	Char-Code BB		Chest R Bottom	/
00002CF0	70-77	Char-Code BC			Parakeet L Top
00002CF8	78-7F	Char-Code BD		(BC - BE Blank)	Parakeet R Bottom
00002D00	80-87	Char-Code BE			
00002D08	88-8F	Char-Code BF		Orb R Top	
	Bytes	Definable Graphic space, Color set by	Byte 88	**Quest** = 1A	**Pennies** = 1B
00002D10	90-97	Char-Code C0			Top L Bottom
00002D18	98-9F	Char-Code C1			Top R Bottom
00002D20	A0-A7	Char-Code C2			Stardust L Bottom
00002D28	A8-AF	Char-Code C3		Lantern Bottom	Stardust R Bottom
00002D30	B0-B7	Char-Code C4		Gold	Magic Glass L Bottom
00002D38	B8-BF	Char-Code C5		King's Face L Side	
00002D40	C0-C7	Char-Code C6		King's Face R Side	
00002D48	C8-CF	Char-Code C7		Orb L Bottom	
	Bytes	Definable Graphic space, Color set by	Byte 89	**Quest** = 16	**Pennies** = DE
00002D50	D0-D7	Char-Code C8		\	
00002D58	D8-DF	Char-Code C9		\ Touchstone	Parakeet
00002D60	E0-E7	Char-Code CA		/	
00002D68	E8-EF	Char-Code CB		/	
00002D70	F0-F7	Char-Code CC		King's Crown L Side	
00002D78	F8-FF	Char-Code CD		King's Crown R Side	
Game	Sector 004F - *Graphic Character Space for Room Contents*				
Byte #	Bytes				
00002D80	00-07	Char-Code CE			
00002D88	08-0F	Char-Code CF		Orb L Top	
	Bytes	Definable Graphic space, Color set by	Byte 8A	**Quest** = 16	**Pennies** = 16
00002D90	10-17	Char-Code D0		Chest L Top	\
00002D98	18-1F	Char-Code D1		Chest R Top	\
00002DA0	20-27	Char-Code D2		Lantern Top	\ Same as
00002DA8	28-2F	Char-Code D3		Potion Top	/ Quest
00002DB0	30-37	Char-Code D4		Dead Thing L Side	/
00002DB8	38-3F	Char-Code D5		Dead Thing R Side	/
00002DC0	40-47	Char-Code D6			
00002DC8	48-4F	Char-Code D7			
Game	Sector 004F - *Graphic Character Space for Room Designs* (note #4)				
Byte #	Bytes	**Room Graphics**,	Color set by Byte 8B (**Cannot be relocated**)		**Quest** & **Pennies** = 1E
00002DD0	50-57	Char-Code D8	Vertical Room Wall		
00002DD8	58-5F	Char-Code D9	Horizontal Room Wall		
00002DE0	60-67	Char-Code DA	Vertical Room Doorway		
00002DE8	68-6F	Char-Code DB	Horizontal Room Doorway		
00002DF0	70-77	Char-Code DC	Room Corner Post		
00002DF8	78-7F	Char-Code DD	Grid Pattern outside a Room		
00002E00	80-87	Char-Code DE	Horizontal Room Jct.	(not actually used, free - note #5)	
00002E08	88-8F	Char-Code DF	Vertical Room Jct.	(not actually used, free)	
	Bytes	Definable Graphic space, Color set by	Byte 8C	**Quest** = 16	**Pennies** = 16
00002E10	90-97	Char-Code E0		Sword L Side	\
00002E18	98-9F	Char-Code E1		Sword R Side	\ Same as
00002E20	A0-A7	Char-Code E2		Bow L Side	/ Quest
00002E28	A8-AF	Char-Code E3		Bow R Side	/
00002E30	B0-B7	Char-Code E4		(Char Codes E4 – EF are Blank)	
00002E38	B8-BF	Char-Code E5			
00002E40	C0-C7	Char-Code E6			
00002E48	C8-CF	Char-Code E7			

V. Graphics in the TOD Environment
Graphics Bank #2 - Room Design & Contents

Game	Sector 004F - *Graphic Character Space for Room Contents*			
Byte #	Bytes	<u>Definable Graphic space</u>, Color set by <u>Byte 8D</u>	**<u>Quest</u>** = <u>00</u>	**<u>Pennies</u>** = <u>00</u> (note #5)
00002E50	**D0-D7**	Char-Code E8		
00002E58	**D8-DF**	Char-Code E9		
00002E60	**E0-E7**	Char-Code EA		
00002E68	**ED-EF**	Char-Code EB		
00002E70	**F0-F7**	Char-Code EC		
00002E78	**F8-FF**	Char-Code ED		
	Sector 0050 - *Graphic Character Space for Room Contents*			
00002E80	**00-07**	Char-Code EE		
00002E88	**08-0F**	Char-Code EF		

Notes:

#1 Stairs, Game Logo, Vault and Room Design Graphics cannot be assigned to new memory locations.

#2 However, if you do not use Vaults the memory space can be reassigned for other graphic use as seen in ***Pennies***. ***Pennies*** is a modified ***Quest*** database as witnessed by the residual ***Quest*** defined graphics codes in ***Pennies***.

#3 The numbers in parentheses illustrates how "Stores & Vaults" graphics are Accessed in a ***Quest*** type game base.

#4 Graphics for the <u>Room View</u> as contrasted with the <u>Hallway view</u>. I.e. anytime the view changes to an overhead view.

#5 *Character-codes* **DE** & **DF** are defined in ***Quest*** and ***Pennies***, but not actually used by the Game Program (Module). These *character-codes* may be redefined for use in a game base.

For the Ranged and Magical Weapon sequences, as well as for the majority of items in Bank #2, an alternate and more flexible method is available. A number of locations are used to store the *character-code* definitions of <u>specific items</u> so that greater creativity in graphics/ color combinations can result. In each instance a total of <u>4 *character-codes*</u>, generated by a 64 hex digit *pattern-identifier*, are necessary to define each item:

The *first character-code* defines the <u>left top quarter</u>, the *second* the <u>left bottom quarter</u>, the *third* the <u>right top quarter</u> and the *fourth* the <u>right bottom quarter</u> of the graphic. (See the ***Extended Basic*** manual, pages 56 to 58 for additional details.) If a graphic requires less than 4 *character-codes* for definition, then code **7B** is used for the remaining two or three codes. This places the "room floor/ map background" color and texture as a filler.

Char-Code Locations to define Room Items

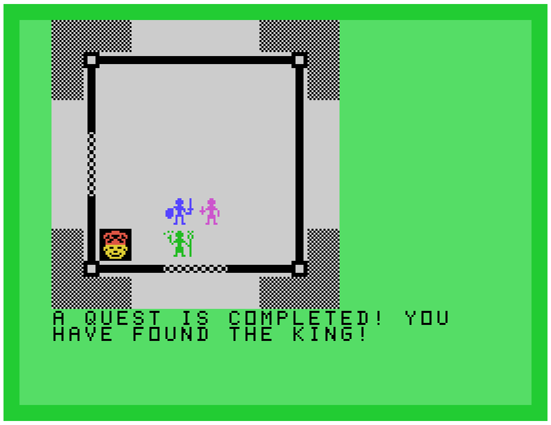
Game	Sector 0050		
Byte #	Bytes	Place:	
00002E90	**10-13**	*Character-Codes* for Room Fountains	\
00002E94	**14-17**	*Character-Codes* for Living Statues	\
00002E98	**18-1B**	*Character-Codes* for Hand Weapon	\ These definitions <u>must all be four</u>
00002E9C	**1C-1F**	*Character-Codes* for Ranged Weapon	\ <u>*char-codes* each</u>.
00002EA0	**20-23**	*Character-Codes* for Shield	\
00002EA4	**24-27**	*Character-Codes* for Armor	/
00002EA8	**28-2B**	*Character-Codes* for Chest	/ Use *char-code* <u>7B</u> to place
00002EAC	**2C-2F**	*Character-Codes* for Gold	/ a blank code as fill (to match floor).
00002EB0	**30-33**	*Character-Codes* for Map	/
00002EB4	**34-37**	*Character-Codes* for Dead Thing	/

Other Graphics Bank #2 Character-Code Locations

➢ 8 Quest Items	(<u>Sector 004B</u>, the 3rd-6th Attribute bytes)
➢ 8 Categories of Magical Items	(<u>Sector 0047</u>, first 4 Attribute bytes)

To repeat, the significance of allowing any *character-code* to be utilized for ***any*** specific graphic is that it permits the full *foreground/ background* coloring scheme, as implemented in the ***TI Basic***/ ***Extended Basic*** **CALL COLOR** subprogram, for use in ***Tunnels of Doom***.

VI. Graphics in the TOD Environment
Graphics Bank #2 - Examples



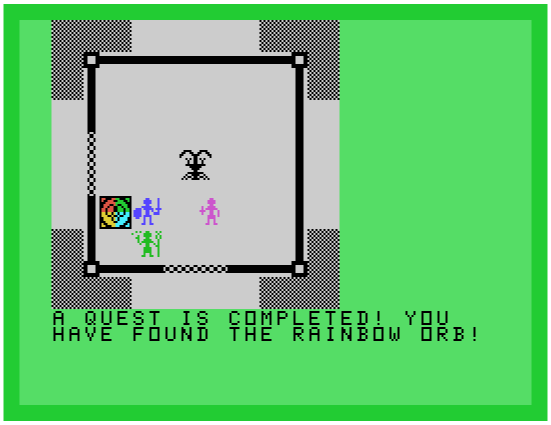
```
A QUEST IS COMPLETED! YOU
HAVE FOUND THE KING!
```

Example from Quest - The King

```
Game         Sector 004B
Byte #       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002980     0 00 00 32 3A 3E 46 5A 5F 5F 64 00 00 3C 44 48 50
00002990     1 5A 5F 5F 64 00 00 4B 49 4E 47 20 20 20 20 20 20
000029A0     2 20 02 0A CC C5 CD C6 00 0C 52 41 49 4E 42 4F 57
000029B0     3 20 4F 52 42 00 0A CF C7 BF B7 00 0E 20 20 20 20
```

`4B 49 4E 47` = *character-codes* used to spell “KING”

`CC C5 CD C6` = *character-codes* used to define the graphic representation of the <u>King</u>



```
A QUEST IS COMPLETED! YOU
HAVE FOUND THE RAINBOW ORB!
```

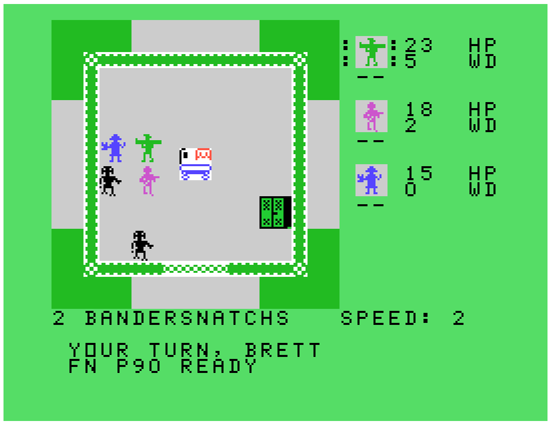
Example from Quest - Rainbow Orb

`52 41 49 4E 42 4F 57 20 4F 52 42` = “RAINBOW ORB” (20=space)

`CF C7 BF B7` = *character-codes* used to define the <u>Rainbow Orb</u>

The *character-codes* are defined in this order: First = top L, Second = lower L, Third = top R, Fourth = lower R of a graphic. (As is done in TI Basic and Extended Basic.)

If fewer than 4 *character-codes* are necessary to define a graphic, then *character-code* **7B** is used to meet the 4 code requirement.



```
2 BANDERSNATCHS    SPEED: 2

 YOUR TURN, BRETT
 FN P90 READY
```

Example from Sub12 - Med Station

Char-Code Locations to define Room Items

```
Game          Sector 0050
Byte #        Bytes    Character-Codes Locations for:
00002E90      10-13    Room Fountains (Med Station)
00002E94      14-17    Living Statues
00002E98      18-1B    Hand Weapon
00002E9C      1C-1F    Ranged Weapon

00002E90 D2 80 E9 81 82 CD 83 CF A4 7B A5 7B BE 7B BF 7B
00002EA0 AC AD AE AF A8 A9 AA AB BA BB BC BD C4 7B 7B 7B
00002EB0 B0 B1 B2 B3 99 A6 9A A7 01 03 00 01 06 42 43 46
```

<u>Remember</u>: If fewer than 4 *character-codes* are necessary to define a graphic, then *character-code* **7B** is used to meet the 4 code requirement.

VI. Graphics in the TOD Environment
Graphics Bank #2 - Examples

Graphics Summary for Rooms & their Contents

Character-codes **Required**:

Definable Locations: Room Items = 40, Quest Items = 32, Magical Items = 32.	104
Fixed Locations: Stairs Up/ Down = 8, Logo = 4, Vault = 4, Rooms = 6.	22
Total potentially necessary *character-codes*:	126

This total ***does not*** include any *character-codes* which may be necessary to embed graphics into text.

Character-codes **Available**:

Graphic Bank #2 Space (**80-EF** or 16 x 7)	112
Unused Map Codes (max) - must be Map colors	4
Unused FC-FF codes - keep red on gray (**6E**)	4
Total potentially available *character-codes:*	120

In total there are 112 *character-codes* routinely available for use by Bank #2, out of the 126, which may be necessary for a game design. This shortfall is solved by some graphics using only 1 (e.g. gold) or 2 (e.g. hand or ranged weapons, dead thing) defined *character-codes*. The remaining (2 or 3) *character-codes* are filled by *character-code* **7B**, which is the design (and color) of the room floor and the map background (generally blank or "apparently empty"). Alternately, certain items can be omitted from use, e.g. chests, vaults or statues. For every Player Character **not** used in a game base, an additional 8 graphic *character-codes* become available.

VII. Graphics in the TOD Environment

Graphics Bank #3 - Compass Directions, Cursors, Ranged & Magical Attacks

Graphic Bank #3 utilizes a mix of definitions for its graphics. Fixed memory locations are used to define the hex strings for compass directions, the map and ranged attack cursor graphics, but *character-codes* are used to link to graphic strings for the actual Ranged and Magical attack sequences. The colors for the statically defined locations appear to be controlled by the module. The "ammo" of the ranged weapon and magical attack sequences assumes the color of the character (Player) that initiated the attack sequence.

Compass Directions, Map Location & Weapon Cursor Graphics (fixed locations)

Game Byte #	Sector 0022 Bytes		Char-Codes
00000080	**00-1F**	Graphic for 'N'orth	80-83
000000A0	**20-3F**	Graphic for 'E'ast	84-87
000000C0	**40-5F**	Graphic for 'S'outh	88-8B
000000E0	**60-7F**	Graphic for 'W'est	8D-8F
00000100	**80-9F**	Party location indicator on map	90-93
00000120	**A0-BF**	Ranged Weapon/ Magical cursor graphic	94-97

Ranged Weapon & Magical Attack Graphic Sequences (Codes 98-BF) (as defined in ***Quest***)

00000140	**C0-DF**	Flying blade graphic #3 (***invoked by spells #70 & 71***)	98-9B
00000160	**E0-FF**	Flying blade graphic #2	9C-9F

Game Byte #	Sector 0023 Bytes		
00000180	**00-1F**	Flying blade graphic #1	A0-A3
000001A0	**20-3F**	Flying blade graphic #4	A4-A7
000001C0	**40-5F**	Ranged weapon Projectile graphic #1 & 3	A8-AB
000001E0	**60-7F**	Ranged Weapon Projectile graphic #2	AC-AF
00000200	**80-9F**	Ranged Weapon Initial Effect & Projectile graphic #4	B0-B3
00000220	**A0-BF**	Magical Weapon Initial Effect & Impact for Ranged Attack	B4-B7
00000240	**C0-DF**	Impact from Magical Weapon	B8-BB
00000260	**E0-FF**	(Blank)	BC-BF

Note: The memory used to define Char-Codes 98 - BF is completely re-definable for both Ranged and Magical Weapons sequences. See Sector 004B, Bytes AE - BD for allocating details. All graphics defined in Bank #3 use 4 *character-codes* each.

Bank #3:

- Ranged Weapon sequence (Sector 004B, bytes AE-B5)
- Magical Attack sequence (Sector 004B, bytes B6-BD)

The Ranged Weapon and Magical attack sequences make use of sprite graphics as implemented in TI Extended Basic. Both of these attacks can use any *character-code* definition in the **98-BF** range (of Bank #3). Their 8 bytes define:

Game Byte #	**Sector 004B, Bytes:**	
00002A2E	**AE-B5**	***Character-Codes* for Ranged Weapon Sequence** (AE - B1=Ordinance Sequence, B2=Impact Effect, B3=Effect offset, B4=Initial Effect, B5=Ending Graphic e.g. 20 or Space)
00002A36	**B6-BD**	***Character-Codes* for Magical Weapon Sequence** (B6 - B9=Ordinance Sequence, BA=Impact Effect, BB=Effect offset, BC=Initial Effect, BD=Ending Graphic e.g. 20 or Space)

Note: The Initial Effects do not replace, but produce an overlay over/ about/ near the Defensive graphic of the character for Magical Attacks, or the Attack graphic for Ranged Weapon attacks . The Impact Effect appears if you successfully hit the target.

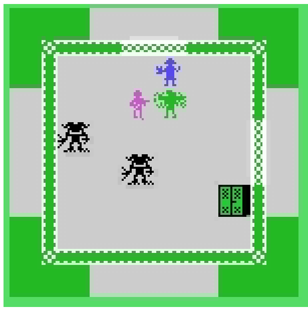
For all graphics represented in the Ranged and Magical Weapon sequences, only the first *character-code*, the one that indicates the L upper quarter of the graphic, is represented in the sequence. **TOD** automatically uses the following 3 graphic codes as part of the definition. In this respect **TOD** functions precisely like the **Extended Basic** version of the **Call Character** subprogram, which can also define and use up to 4 *character-codes* at a time depending on the hex string length of the *pattern-identifier*.

The Ranged and Magical Weapon attack sequences used in ***Sub12***:

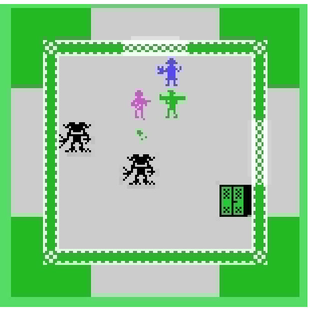
```
Game       Sector 004B from Sub12
Byte #      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002A20 A 20 20 20 20 20 20 00 00 00 00 00 00 00 00 A8 A8 Ranged weapon
00002A30 B A8 A8 B4 04 B0 20 A0 A0 A0 A0 B8 05 BC 20 4F 50 SMAW Attack
```

Note that ***Sub12*** uses a single graphic code for ordinance, `A8` for ranged and `A0` for magical.

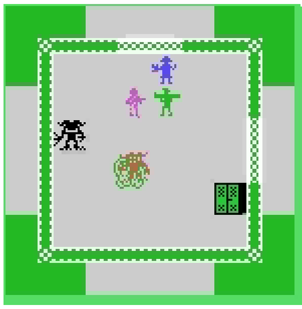
Example of Magical Weapon Sequence - from Sub12



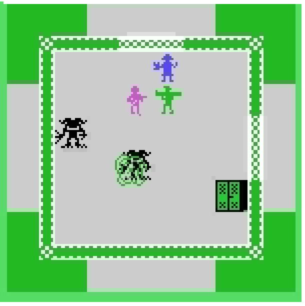
#1 Byte BC - code BC



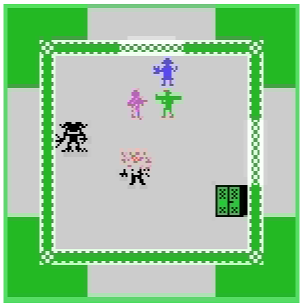
#2 *Bytes* B6-B9 - code <u>A0</u>



#3 *Bytes* BA-BB-code B8 & hex 05



#4 *Bytes* BA-BB



#5 Dead-Thing

You may need to increase the Zoom Magnification to view details.

#1 *Byte* `BC` stores the *character-code* for a brief overlay graphic used to present the <u>initial effect</u> from the use of a Magical Weapon. Here, it is represented by a "smoke plume" from the firing of the SMAW carried by Brett (the figure in green).

#2 *Bytes* `B6-B9` store the *character-codes* that graphically represent the <u>ordinance</u> in flight; be it an arrow, fireball, RPG, or the flying blade used in Quest. The ordinance can be a single graphic as seen above, or have up to 4 varying forms while on route. For a single view all four bytes are populated with the same *character-code*.

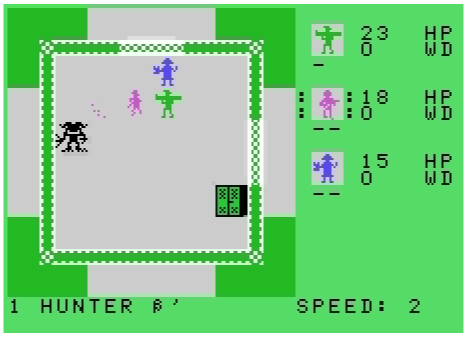
#3 Illustrates the <u>impact effect</u> of your weapon, which is defined in *byte* `BA` and the impact effect <u>offset</u> defined in *byte* `BB`. The impact effect is another overlay graphic. The initial and impact overlays, as well as the ordinance graphics are actually sprites, which explain how these effects are produced. Note that on successful impact the monster changes to its attack graphic form and color. Also note how both the effects and ordinance take on the color of the character that initiated the attack. The <u>offset</u> is a hex digit that exaggerates the impact effect in the direction opposite to the point of impact - the larger the number, the greater the "exaggeration". The flying blade effect in ***Quest*** uses `01` for its offset, because the blade surrounds and cuts through its victim. The example above from ***Sub12*** uses `05` as its offset to mimic an explosive type of impact.

#4 The monster returns to its defense graphic and color, which generally enables a better view of the impact effect.

#5 If the attack was fatal, the monster graphic is replaced with the graphic representation for a Dead-Thing defined by the *character-codes* in `Game Bytes  #00002EB4-00002EB7` (`Sector 0050, Bytes 34-37`) before it is removed.

The bytes reserved for Ranged and Magical Attacks work identically in both instances. The final *byte* in the sequences, `B5` for Ranged and `BD` for Magical, represent the <u>ending graphic</u> code (`hex 20` or space character), which terminates the ordinance. If this is not employed the ordinance will scroll beyond one edge of the screen to the other, several times, before deleting. It appears to be the equivalent of the **CALL PATTERN** subprogram used in Extended Basic, where in this instance the replacement sprite is the space character.

The overlays and ordinance sequence for ranged weapon attacks are less spectacular in their presentation. Due to their smaller size and briefness they are also more difficult to successfully display. Remember, only a single graphic representation sequence is available for ordinance type.

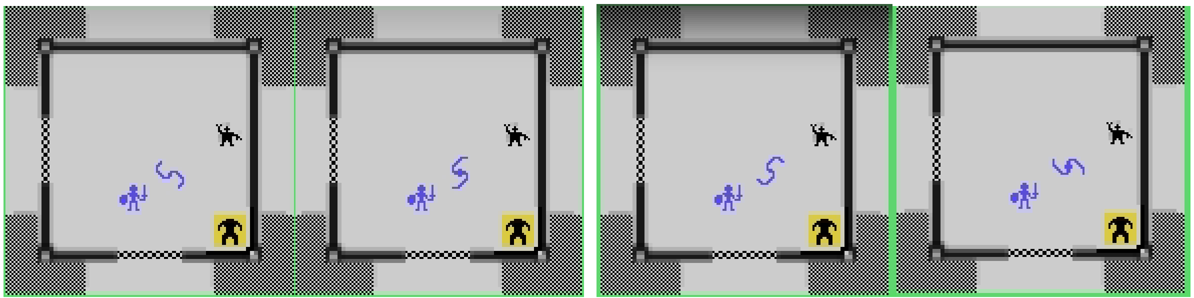


To mix it up a bit ***Sub12*** used a representation of a 5 shot burst for ranged ordinance. As most of the ranged weapons were automatic, semi-automatic or shotguns, this seemed to be an appropriate display overall!

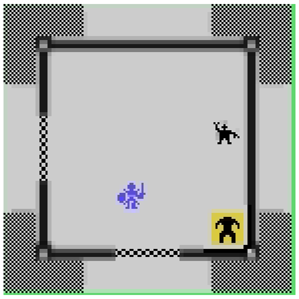
Left: Dex, the magenta character, firing a 5-shot burst.

```
Game      Sector 004B from Quest
Byte #     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002A20  20 20 20 20 20 20 00 00 00 00 00 00 00 00 A8 AC Ranged Weapon
00002A30  A8 B0 B4 04 B0 20 A0 9C 98 A4 B8 01 B4 20 4F 50 Magical Attack
```

The extraordinary flying blade sequence used in ***Quest*** demands review. It is a bit more complicated, using 4 unique character-codes for its ordinance sequence:

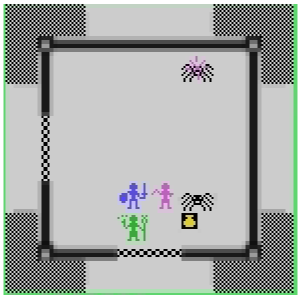


Character-codes: A0 9C 98 A4



The Magical Weapon impact effect used in ***Quest*** is the same as shown previously for ***Sub12***, but the <u>offset</u> is **hex 01** thus the effect is centered directly on the monster.

For the Magical Weapon <u>Initial</u> effect graphic, ***Quest*** uses code **B4**, which is <u>also</u> the <u>Impact</u> effect graphic it uses for a Ranged Weapon attack. The <u>Initial</u> effect is more difficult to see being blue on blue, while the <u>Impact</u> is mauve on black.



Quest - Magical Weapon <u>Initial</u> Effect

Quest - Ranged Weapon <u>Impact</u> Effect

Game Title and Description (First 8 lines of text, 32 columns each)

```
Game      Sector 0024
Byte #    Bytes
00000280  00-1F  1st line of text/ graphics (i.e. Game Title)
000002A0  20-3F  2nd line of text
000002C0  40-5F  3rd line of text
000002E0  60-7F  4th line of text
00000300  80-9F  5th line of text
00000320  A0-BF  6th line of text
00000340  C0-DF  7th line of text
00000360  E0-FF  8th line of text
```

Game Title and Description (Last 4 lines of text, 32 columns each)

```
Game      Sector 0025
Byte #    Bytes
00000380  00-1F  9th line of text/ graphics
000003A0  20-3F  10th line of text
000003C0  40-5F  11th line of text
000003E9  60-7F  12th line of text
00000400  80-9F  Not used \
00000420  A0-BF  Not used  \ You may place Game developer information, date of
00000440  C0-DF  Not used  / completion, game version #, etc. here as a type of
00000460  E0-FF  Not used /  REM-Statements.
```

Note: The first 8 lines of the Game Description Screen are reserved by the module for the ***'Tunnels of Doom'*** title and status reports on constructing/ stocking of new dungeons. The module accepts 12 lines of text, 32 columns each of any character in the 00 - 5F range. The first line of text is typically used for the Game Title and this is what is displayed at the end of a game if you have successfully completed the same, along with any Quest objects you successfully recovered. ***Halls*** places graphics on the margins of this text, ***Fritz's Editor*** places his software logo here.

Char-Codes 00-1F are the Defense and Attack graphics of up to 4 Characters, listed in the order that they were selected for play from the last saved game. (Also available if "Continue Current Game" is selected.) 20-5F are the same as those from Sectors 0027 & 0028. Above Char-Code 5F colored square blocks are primarily available. Orange from 60-6F. Green 70-77. Blue 78-7F. Magenta 80-87. Red 88-E7. Typically A0 is used to provide the colored border. Codes F0-F3 will display the Defense graphic and F8-FB the Attack graphic of the last Monster encountered in a saved game. F4-F7 the stairs down of saved game. E8-EF Blue messy graphics. If no monsters were encountered before the game was save, F0-F7 displays yellow squares. Bytes 80-FF of Sector 0025 are unused, but no additional text is accepted. It can be used as a **REM** area for Game Developer information, version etc.

VIII. A List of the World's Known Spells . . .
. . . recognized by Tunnels Of Doom

An understanding of how TOD manipulates graphic *character-codes* provides a suitable foundation for understanding the concept of Spells and Magical effects used within the game. For efficiency of memory use and convenience, all "effects" are cataloged by a single byte of data, again represented by 2 hex digits. These spells or effects factor into a number of the Game Lists, which form a significant part of every game database. These Lists include:

- Monster Special Attacks (20 effects)
- Magical Item Effects (40 effects)
- Fountain Effects (10 effects)
- Chest Traps (10 effects)
- Miscellaneous effects: these can either increase or decrease Player Damage (Ration Consumption, Stores, Vaults, etc.

Regardless of the actual spell or effect, in most instances (except for Fountains) the Spell or Item itself can be given a specific name e.g. Corrosion, Healing Light, Honing Stone, etc. However, the name assigned may or may not (at the discretion of the game author) have relevance to the effect called forth! In addition, any effect *regardless of its source*, e.g. the 20 types of effects specified in the list of Monster Special Attacks, can be beneficial or detrimental to the player!

The Lists that incorporate the various Spells are individualized in their treatment. In some instances detailed duration, or number of uses; and intensity, or amount are provided. In other situations only the Spell itself can be specified and the consequences of the spell are furnished by the TOD program. Whether the details of a Spell are furnished by the game designer, or derived from a routine within the program, they all share the same characteristics.

The Three Characteristics of every Spell or Magical Item

The first characteristic of every Spell or Magical Item is: specific effect. This is always the responsibility of the game developer except for a few effects that only increase (or decrease) *Player Damage*. A Spell needs to be specified for use, or to become part of a random 'Spell Bank'. All spells are drawn from the TOD's "List of Known Spells". The programmer of the TOD program devised an ingenious scheme whereas every even numbered Spell *increases* a specific effect, while the immediately following odd number *decreases* the same effect.

The second characteristic of every Spell or Item is: number of uses. This may indicate an absolute value such as "3 times", or it may represent duration in key presses (paces). A `hex 03` as duration would indicate 30 paces. Magical objects can frequently be used more than once. Fountains and Chest Traps trigger an immediate effect at the time of contact (though the same effect can recur multiple times!)

The third factor is: intensity (amount for good or ill). Most Spells add or subtract from the Party's, a Player's, or the Monsters' stats: Hit Points, Experience, Armor Bonus, etc. Only the last three Spells – regulated by the game program express neither intensity nor duration, but are either "on or off": *Shows Monsters in Room*, *Finds Traps*, and *Makes Traps Misfire*.

For our current purpose, we will not delve into the various Lists that utilize the various magical effects but focus solely on the effects of the spells themselves and their broad, general classifications. The most general classification of spells organizes them into three categories. Those that effect: a specific Player, the entire Party, or Monsters and Traps.

VIII. A List of the World's Known Spells . . .
. . . recognized by Tunnels Of Doom

The tabulation of effects generated by the various spells are arranged and classified under three categories:

Hex#	**Effects: Player**
00	↑ Player HP
01	↓ Player HP
02	
03	
04	↑ Player Damage
05	↓ Player Damage
06	
07	
08	↑ Player Armor Protection
09	↓ Player Armor Protection
0A	
0B	
0C	↑ Player Weapon Damage
0D	↓ Player Weapon Damage
0E	
0F	
10	↑ Player Armor Bonus
11	↓ Player Armor Bonus
12	
13	
14	↑ Player Weapon Bonus
15	↓ Player Weapon Bonus
16	
17	
18	↑ Player Luck
19	↓ Player Luck
1A	
1B	
1C	↑ Player Experience
1D	↓ Player Experience
1E	
1F	
20	↑ Player Level
21	↓ Player Level
22	
23	
24	(Not Used)
25	(Not Used)
26	
27	

Hex#	**Effects: Party**
28	↑ Party Damage
29	↓ Party Damage
2A	
2B	
2C	↑ Party Gold
2D	↓ Party Gold
2E	
2F	
30	↑ Party Rations
31	↓ Party Rations
32	
33	
34	↑ Party Weapon Availability
35	↓ Party Weapon Availability
36	
37	
38	(Not Used)
39	(Not Used)
3A	
3B	
3C	↑ Party Combat Speed
3D	↓ Party Combat Speed
3E	
3F	
40	↑ Wandering Monster Probability
41	↓ Wandering Monster Probability
42	
43	
44	↑ Party Ration Consumption Time
45	↓ Party Ration Consumption Time
46	
47	
48	↑ Party Healing Interval
49	↓ Party Healing Interval
4A	
4B	

Hex#	**Effects: Monsters or Traps**
4C	↑ Monster Armor Protection
4D	↓ Monster Armor Protection
4E	
4F	
50	↑ Monster Attack Class
51	↓ Monster Attack Class
52	
53	
54	↑ Monster Attack Damage
55	↓ Monster Attack Damage
56	
57	
58	↑ Monster Special Power Chance
59	↓ Monster Special Power Chance
5A	
5B	
5C	↑ Monster Bribability
5D	↓ Monster Bribability
5E	
5F	
60	↑ Monster Mobility
61	↓ Monster Mobility
62	
63	
64	↑ Monster Magical Resistance
65	↓ Monster Magical Resistance
66	
67	
68	↑ Monster Combat Speed
69	↓ Monster Combat Speed
6A	
6B	
6C	↑ All Monster Hit Points
6D	↓ All Monster Hit Points
6E	
6F	
70	↑ Monster Hit Points
71	↓ Monster Hit Points
72	
73	
74	Show Monsters in Room
75	Show Monsters (Not used)
76	
77	
78	Finds Traps
79	Makes Traps Misfire
7A	
7B	
7C	(Not Used)
7D	(Not Used)
7E	
7F	

The hex code invokes the spell/ effect.
Even numbers *Increase*, odd numbers *Decrease* effects in the Hex 00-71 range.
The next two numbers repeat the same spell/ effect.
6C & 6D invoke the Yellow background color on all monsters.
70 & 71 initiate the Flying Blade or Magical Weapon sequence.
Highlighted spells are expressed in duration of time.

VIII. A List of the World's Known Spells . . .
. . . recognized by Tunnels Of Doom

Some general observations may prove helpful. All spells are grouped into four *"character-codes"* like sets similar to their graphical counterparts. The sequence appears to end just before *hex* 80 and this is probably not coincidental! It appears that spell numbers are an equivalent of the common graphic Bank of codes 00-7F in terms of implementation. However, this is conjecture.

It was mentioned that an even hex number increases an effect and the immediately following odd hex number decreases the same effect. The two unused hex codes that follow appear to repeat the same result or effect - even increasing and odd decreasing. The very fact that a spell number is even or odd triggers the "Increases" or "Decreases" dialogue. While a spells effect is always specific, its number of uses and intensity may be variable or precise.

There are four Spells affecting a player's Party, which are expressed in duration of time (paces or key presses) instead of a number of uses; *Party Combat Speed* (02), *Wandering Monster Probability* (03), *Party Ration Consumption Interval* (02) and *Party Healing Interval* (02). These duration spells always affect the Party as a whole and never just an individual member. Spells that affect Monsters, except for Increasing or Decreasing *Monster Hit Points*, affect all Monsters. (Default ***Quest*** settings in parenthesis).

Any ordinary hex number used to express uses or intensity represents the maximum number of uses or amount for that specific spell. This technique produces a sense of game randomness, which is a very important factor in maintaining player interest and game variability. However, by expressing these values in reverse notation, or negative numbers, a spell's number of uses and/or intensity will always be the same.

For example, ***Sub12*** assigns an SMAW II Serpent to Brett at the start of the game. In the list of 40 Spells it is defined in Sector 004A as:

Spell #		Name Bytes		Effect Bytes	Spell	Duration or Uses	Amount or Intensity	
1E		1E-2C	SMAW II SERPENT	2D-2F	71	FA	64	↓ Hostile HP

From the list of Known Spells we see that #71 *Decreases Monster Hit Points*. The number of uses is FA, which means this spell can be used *exactly* 6 (hex FA=06) times. Had hex 06 been placed here it would have indicated a random 1 to 6 uses. The intensity or amount of damage with each use will be up to 100 Hit Points (hex 64=100). Had the last been written as E2, i.e. in reverse notation, the amount of damage would have been 30 with every use of this spell. All spells can be precisely defined in this manner, but overuse is at the expense of any pretense of game variability or "player luck". For the majority of spells this technique should be used sparingly, however exceptions do exist.

Spells that express the number of uses as duration in key presses (paces) are good candidates for the use of reverse notation values. Duration values have default settings in a game database. A spell that affects duration is added to, or subtracted from this baseline value. Examples from Quest:

Spell #		Name Bytes		Effect Bytes	Spell	Duration or Uses	Amount or Intensity	
6	**Lantern**	6E-7C	Soothing Light	7D-7F	49	FA	FE	Dec. Party Healing Interval
7		80-8E	Aura of Warding	8F-91	41	FA	FF	Dec. Wandering Monster Prob.
8		92-A0	Rainbow Strobe	A1-A3	29	02	05	Dec. party Damage
9		A4-B2	Path Lightener	B3-B5	44	FA	FC	Inc. Ration Consumption Interval
A		B6-C4	Smudge Pot	C5-C7	40	FA	0A	Inc. Wandering Monster Prob.

13		58-66	Repair	67-69	08	03	FE	Inc. Player Armor Protection

FA for spell uses (duration) defines that these spells will remain in effect for 60 paces FA=06 (x A or 10) each. After 60 paces they will return to their baseline time values. FE as amount in Spell #06 designates Healing occurring every: baseline-20 paces FE = 02 (x A or 10). As the duration value cannot be less than 01, or every 10 paces, Quest lowers the Healing Interval to every 10 paces - assuming the baseline had not been temporarily elevated by another spell! The Wandering Monster Probability in Spell #07 is decreased from a 30% (hex 03) baseline value by FF = 01 or 10%, to 20%. FC Decreases the Ration Consumption from every (02) = 20 paces (baseline) to every 20 + FC = 04 (40) = 60 paces, decreasing consumption by 1/3 for 60 paces. Spell #A Increases Wandering Monster Probability by a random value up to 100%. Our last example will Increase Player Armor Protection by exactly 2 points from its current value. When a spell's duration is expressed in key presses the possibility exists that an effect opposite to that planned may occur as a random number may end up being above or below an established baseline. In such instances a specific amount or duration is desirable to clearly define the spell as beneficial or detrimental with every use, and to make the spell's end result reasonable.

Spell codes 74 and above are treated somewhat differently. First, there is no *Increases* or D*ecreases* dialogue initiated with their use. Secondly, the dialogue phrase that describes their effect is controlled entirely by the command module whereas all previous dialogues may be modified by the programmer. Last of all, their effects are *absolute* and not evaluated in terms of points, duration, or quantity like those of spells 00-73.

When a new Magical Item is found within a dungeon, the Spell number is initially defined in reverse notation, which flags the TOD program that the owner is unaware of its specific properties. This produces the "UNTRIED POTION", or whatever the category name of the spell, dialogue. The specific properties of a Magical Item are revealed when the item is first used, or this information is obtained from a Living Statue. In either of these two instances the Spell number reverts to positive hex digits; the owner can now see and obtain information on its properties when viewing it in a Player's Inventory.

The Spell Table is internal to the Tunnels of Doom programming and does not itself directly appear in any game database list. A few additional notes on specific spells:

- Spells that influence weapon or armor Damage or Protection are infused into the item proper and the benefit/ detriment to the Player vanish if that item is discarded.
- Spells that influence a Player's Bonus become a permanent benefit or detriment to the Player.
- Increasing or Decreasing Weapon Availability enables a weapon that can typically only be used once per battle to be used more than once (or not at all).
- Increasing or Decreasing Combat Speed gives the Player or Monster an additional opportunity to inflict damage. Care must be taken when assigning an Increased Combat Speed to Monsters lest "Game Over" happens well before you intended. Generally the more powerful the Monster the less you increase their Combat Speed or make it a very *small probability*.
- Increasing or Decreasing All Monster Hit Points (#6C & 6D) invokes the Yellow background effect around Monsters.
- Increasing or Decreasing Monster Hit Points (#70 & 71) invokes the Magical Attack sequence; i.e. the Flying Blade effect used in ***Quest***, the Fireball effect in ***Halls*** and the RPG effect used in ***Sub12***.

The dialogue generated with spell usage may be modified for spells in the #00-71 range. These are manipulated through modification of a table found in Sector 0052, bytes 40-9F, which follows.

VIII. A List of the World's Known Spells . . .
. . . recognized by Tunnels Of Doom

List of the World's Known Spells Recognized by the TOD Module (from Quest)

Game	Sector 0052		Dialogue for	
Byte #	**Bytes**	**Code**	**Spells**	**Effect**
000030C0	**40-42**	305520	00 & 01	PLAYER HIT POINTS
000030C3	**43-45**	304C20	04 & 05	PLAYER DAMAGE
000030C6	**46-48**	303647	08 & 09	PLAYER ARMOR PROTECTION
000030C9	**49-4B**	30354C	0C & 0D	PLAYER WEAPON DAMAGE
000030CC	**4C-4E**	303659	10 & 11	PLAYER ARMOR BONUS
000030CF	**4F-51**	303559	14 & 15	PLAYER WEAPON BONUS
000030D2	**52-54**	305720	18 & 19	PLAYER LUCK
000030D5	**55-57**	303220	1C & 1D	PLAYER EXPERIENCE
000030D8	**58-5A**	303320	20 & 21	PLAYER LEVEL
000030DB	**5B-5D**	2B2B2B	(not used)	(Filled with 2B='+' plus sign symbol)
000030DE	**5E-60**	46304C	28 & 29	ALL PLAYER DAMAGE
000030E1	**61-63**	315420	2C & 2D	PARTY GOLD
000030E4	**64-66**	314120	30 & 31	PARTY RATION
000030E7	**67-69**	31354C	34 & 35	PARTY WEAPON AVAILABILITY
000030EA	**6A-6C**	2B2B2B	(not used)	
000030ED	**6D-6F**	314D43	3C & 3D	PARTY COMBAT SPEED
000030F0	**70-72**	563852	40 & 41	WANDERING MONSTER PROBABILITY
000030F3	**73-75**	41585A	44 & 45	RATION CONSUMPTION INTERVAL
000030F6	**76-78**	31425A	48 & 49	PARTY HEALING INTERVAL
000030F9	**79-7B**	383647	4C & 4D	MONSTER ARMOR PROTECTION
000030FC	**7C-7E**	384B4A	50 & 51	MONSTER ATTACK CLASS
000030FF	**7F-81**	384B4C	54 & 55	MONSTER ATTACK DAMAGE
00003102	**82-84**	384951	58 & 59	MONSTER SPECIAL POWER CHANCE
00003105	**85-87**	385320	5C & 5D	MONSTER BRIBABILITY
00003108	**88-8A**	384820	60 & 61	MONSTER MOBILITY
0000310B	**8B-8D**	384544	64 & 65	MONSTER MAGIC RESISTANCE
0000310E	**8E-90**	384D43	68 & 69	MONSTER COMBAT SPEED
00003111	**91-93**	463855	6C & 6D	ALL MONSTER HIT POINTS
00003114	**94-96**	385520	70 & 71	MONSTER HIT POINTS
00003117	**97-99**	4E3850	74 & 75	ROOM MONSTER INFORMATION (not used)
0000311A	**9A-9C**	2B2B2B	(not used)	
0000311D	**9D-9F**	2B2B2B	(not used)	

Note: The generated dialogue, e.g. "Player Luck" is composed by concatenating the Hex Codes 30 & 57 & 20 together into a phrase (rather like concatenating string variables such as "A$&B$&C$" in TI Basic). 30="Player", 57="Luck" and 20 (Space Character) is used to pad out the string as three Hex Codes are required for each phrase. The "Increase" or "Decrease" dialogue, which precedes the spell phrase occurs automatically, depending whether the Hex Code of the spell is an even (Increases) or odd (Decreases) number.

For "unused spells" the dialogue Hex Code is "2B2B2B', or "+++" in ASCII characters. This is to prevent potentially undesired effects in the database and appears to represent one of several programming bugs in the module. Another is that the module overwrites any attempts to modify the dialogue for spells above Hex #71 (though the original database author attempted to do so! See dialogue for 74 & 75.)

Regardless of how a spell's dialogue phrase is modified, the spell it points to and its effect remain unchanged. So changing bytes 40-42 from "305520" PLAYER & HIT POINTS & (space character) to "30354C" PLAYER & WEAPON & DAMAGE will still increase or decrease the "PLAYER HIT POINTS".

VIII. A List of the World's Known Spells . . .
. . . recognized by Tunnels Of Doom

The 'Code' column in the table is derived from the **Game Key Words** section of the database found in Sectors 0050 - 0052.

Game Key Words

Game	Sector 0050			
Byte #	Bytes	Word	Max. Length	Char-Code
00002ED0	50-5F	FOUNTAIN	18 bytes	
00002EE0	60-6F	LIVING STATUE	16 bytes	
00002EF0	70-7F	GOLD PIECES	16 bytes	
00002F00	80-8F	MAGICAL ITEMS	16 bytes	
00002F10	90-9B	PLAYER	12 bytes	30
00002F1C	9C-A7	PARTY	12 bytes	31
00002F28	A8-B3	EXPERIENCE	12 bytes	32
00002F34	B4-BF	LEVEL	12 bytes	33
00002F40	C0-CB	RANGED	12 bytes	34
00002F4C	CC-D7	WEAPON	12 bytes	35
00002F58	D8-E3	ARMOR	12 bytes	36
00002F64	E4-EF	SHIELD	12 bytes	37
00002F70	F0-FB	MONSTER	12 bytes	38
00002F7C	FC-07	TRAP	12 bytes	39 & 40
	Sector 0051 - *Game Key Words*			
00002F88	08-13	RATION	12 bytes	41 & 3A
00002F94	14-1F	HEALING	12 bytes	42 & 3B
00002FA0	20-2B	SPEED	12 bytes	43 & 3C
00002FAC	2C-37	RESISTANCE	12 bytes	44 & 3D
00002FB8	38-43	MAGIC	12 bytes	45 & 3E
00002FC4	44-4F	ALL	12 bytes	46 & 3F
00002FD0	50-5B	PROTECTION	12 bytes	47
00002FDC	5C-67	MOBILITY	12 bytes	48
00002FE8	68-73	SPECIAL	12 bytes	49
00002FF4	74-7F	CLASS	12 bytes	4A
00003000	80-8B	ATTACK	12 bytes	4B
0000300C	8C-97	DAMAGE	12 bytes	4C
00003018	98-A3	COMBAT	12 bytes	4D
00003024	A4-AF	ROOM	12 bytes	4E
00003030	B0-BB	AVAILABILITY	12 bytes	4F
0000303C	BC-C7	INFORMATION	12 bytes	50
00003048	C8-D3	POWER CHANCE	12 bytes	51
00003054	D4-DF	PROBABILITY	12 bytes	52
00003060	E0-EB	BRIBABILITY	12 bytes	53

Game Key Words - continued

Game	Sector 0051			
Byte #	Bytes	Word	Max. Length	Char-Code
0000306C	EC-F7	GOLD	12 bytes	54
00003078	F8-03	HIT POINTS	12 bytes	55
	Sector 0052 - *Game Key Words*			
00003084	04-0F	WANDERING	12 bytes	56
00003090	10-1B	LUCK	12 bytes	57
0000309C	1C-27	CONSUMPTION	12 bytes	58
000030A8	28-33	BONUS	12 bytes	59
000030B4	34-3F	INTERVAL	12 bytes	5A

Game	Sector 0052		
Byte #	Bytes	Word	
00003120	A0-AB	CHEST	12 Bytes
0000312C	AC-B7	VAULT	12 Bytes

These Key Words are used to generate spell dialogue and contribute to the wording of Status Reports, The General Store, Fountains, Statues, Chests and Vaults. Highlighted Key Words have been identified to have usage in settings other than **Spell Names:** i.e. Status Reports, Help Screens and the General Store.

VIII. A List of the World's Known Spells . . .
. . . recognized by Tunnels Of Doom

It should be *emphatically emphasized* that a thorough understanding of how Spells are devised, used and set forth in the ***Tunnels of Doom*** environment will do precious little to improve your spelling, reading comprehension, dealing with day to day issues, and so forth…. With this matter being thoroughly clarified, we shall now proceed….

Use of a specific hex code causes the corresponding Key Word to be printed on screen and a modification to a number of the Game Key Words requires care. Those highlighted have been identified to have usage in settings other than **Spell Names**, such as ***Status Reports***, ***Help Screens** & the **General Store***; so care must be exercised. Changing the term *Mobility* (hex 48) to *Agility* will cause the word *Agility* to be used in ***Spells*** while the term *Mobility* will remain in use in ***Reports***.

Some Status Report Screens are formatted for specific word lengths (E.g. EXPERIENCE - try substituting WISDOM in its place!). There is a formatting relationship with the words "EXPERIENCE, LEVEL and WEAPON" and another with "ARMOR, SHIELD and PROTECTION". When introducing new terms it is best to keep them the same length as the old term, or add preceding space characters (in some instances). In the example above, if you wish to use the term Wisdom instead of Experience, precede WISDOM with four space characters (Hex 20). Check all help screens and dialogues when any changes are made.

An obvious question would be, "What happens if Codes below hex 30 or above 5A are used?" The answer is that the following is read, 12 bytes at a time.

Codes 01-2F = Nothing except for Char Code 20 = Space
Codes 5B-62 read data from this table beginning at 40 (4D?)
Code 63 = Chest
Code 64 = Vault
Codes 65-94 = Continue to read to the end of the database
Codes 95-FF = Appear to Read from VDP Memory beyond the TOD database - a very scary place...

Specific details on the application of Spell codes used in the various Lists is covered in Chapter IV, regarding Lists, and Chapter VI, which details Monster characteristics.

IX. Making those Lists and Checking them Thrice

Approximately 14 Sectors of each TOD database contain data placed into Lists and for general game settings. To this total we can include another 5 sectors of Monster stats. This chapter will focus on Lists for items and equipment. Specifically those pertaining to:

➢ Weapons	Floor numbers here in reverse notation = All Classes ***may*** use.
➢ Armor	Floor numbers here in reverse notation = All Classes ***may*** use.
➢ Magical Item Categories	Floor numbers here in reverse notation = ***only*** Wizard (special) Class use.
➢ Magical Items	
➢ Quest Objects	
➢ Hallway Fountains	
➢ Chest Traps	

The most practical way to deal with Lists is to List them!

Sectors 0044 – 0046 Weapons List from Quest

List #	Game Byte #	Bytes	Weapon Name	Attribute Bytes	Max Dam	Cost	Floor	Ammo Quantity	Use	Cost	Name Bytes	Ammo Name
Hand Weapons					**A¹**	**A²**	**A³**	**A⁴**	**A⁵**	**A⁶**		
1	00002378	**F8-06**	Dagger	**07-09**	04	01	FF				(0045)	
2	0000238A	**0A-18**	Hand Ax	**19-1B**	06	03	01					
3	0000239C	**1C-2A**	Sword	**2B-2D**	08	05	01					
4	000023AE	**2E-3C**	Wizard Blade	**3D-3F**	08	00	FD					
5	000023C0	**40-4E**	Battle Ax	**4F-51**	0A	08	04					
6	000023D2	**52-60**	Elvin Blade	**61-63**	0C	00	05					
7	000023E4	**64-72**	Dwarven Ax	**73-75**	10	10	06					
8	000023F6	**76-84**	Sword King	**85-87**	14	00	08					
Ranged Weapons												
9	00002408	**88-96**	Sling	**96-9C**	02	01	FF	00	00	00	**9D-A9**	Stones
A	0000242A	**AA-B8**	Short Bow	**B9-BE**	06	03	01	EC	FF	01	**BF-CB**	Arrows
B	0000244C	**CC-DA**	Cross Bow	**DB-E0**	08	06	01	EC	FF	02	**E1-ED**	Quarrels
C	0000246E	**EE-FC**	Alertness Bow **(0046)**	**FD-02**	06	00	04	00	00	00	**03-0F**	Arrows
D	00002490	**10-1E**	Warball & Chain	**1F-24**	14	0F	04	00	FE	00	**25-31**	
E	000024B2	**32-40**	Bow of Strength	**41-46**	0C	00	07	EC	FF	05	**47-53**	Stone Arrows
F	000024D4	**54-62**		**63-68**							**69-75**	
10	000024F6	**76-84**		**85-8A**							**8B-97**	

Note: Weapon, Armor & Shield names can be up to 15 characters long, Ammo names up to 13 characters

Max Damage (A¹) = Maximum amount of Damage the Weapon can inflict with a hit.

Cost (A²) = Cost of the Item (x Factor). Store on Ground Floor = Factor of 10, unless changed. If '00' is placed in this byte = item cannot be purchased, but must be found in dungeon. Examples are the WIZARD BLADE and ALERTNESS BOW in QUEST.

Floor (A³) = Floor on which an Item is first potentially available. E.g. '01' may be found on the 1st-10th floors. If there is a cost listed in **A²**, then it will also be for sale in any Stores on the Ground and lower floors as well. "04' may be found on the 4th and lower floors. It may also be purchased in any 4th Floor Store (or lower Stores), etc.

A '00' placed here means the item can neither be found nor purchased in a Store, but it may be assigned to a Player Class. E.g. the "MARK 30 TASER" in SUB-12. If the floor level is given in reverse notation, e.g. FF, FC, etc. = Wizard Ability Class can use.

Ammo Quantity (A⁴) = Rounds of Ammo that come with weapon. E.g. EC=20 (Use reverse notation). If the number here is '00', or less than the default value in Sector 003A, byte E5, then the E5 value is used. Purchase limit appears to be 120.

Ammo usage (A⁵) = FF=use until out, one/ round. FE=once/ battle, overrides any limit set in A⁴. 00=unlimited ammo. If using 00 or FE in A⁵, set A⁴ to 00. Other reverse notation values behave erratically or freeze the game. If '1' is used, ammo count will not decrease, 2= "one use per combat", but unlimited use, 3 or greater, nothing shows under amount of ammo.

Ammo Cost (A⁶) = Cost of Item x Factor = additional Ammo price in Stores if purchased separately from the weapon.

Weapons are referenced in a game by their hex number (position) in the list (i.e. 1 - 10), regardless of any blank entries present. The Stores however, will only list actual items present beginning with "1" sequentially (1-8 max) and ignore any blank fields.

Sectors 0046 – 0047 Armor List from Quest

List #	Game Byte #	Name Bytes	Armor Name	Attribute Bytes	Prot.	Cost	Floor
Armor					A^1	A^2	A^3
1	00002518	98-A6	Leather	A7-A9	02	02	FF
2	0000252A	AA-B8	Ring Mail	B9-BB	04	05	01
3	0000253C	BC-CA	Plate Mail	CB-CD	06	0A	01
4	0000254E	CE-DC	Elvin Mail	DD-DF	08	00	03
5	00002560	E0-EE	Warrior Mail	EF-F1	09	14	05
6	00002572	F2-00	Hero Mail **(0047)**	01-03	0A	00	07
7	00002584	04-12		13-15			
8	00002596	16-24		25-27			
Shields							
9	000025A8	28-36	Shield	37-39	01	01	01
A	000025BA	3A-48	Magic Armbands	49-4B	02	00	FE
B	000025CC	4C-5A	Dancing Shield	5B-5D	03	00	05
C	000025DE	5E-6C	Cloak of Hiding	6D-6F	05	00	08
D	000025F0	70-7E		7F-81			
E	00002602	82-90		91-93			

Armor & Shield Names - up to 15 characters long. All 14 armor types need not be used and any may be left blank. However, blank entries are still counted in the numbering of Armor. #1-8 are designated as Armor, #9-E as Shields (or whatever you wish to call either).

Protection (A^1) = Maximum amount of Protection.
Cost (A^2) = Cost of the Item (x Factor). Store on Ground Floor = Factor of 10, unless changed.
Floor (A^3) = Floor on which the Item is first potentially available. E.g. '01' may be available from the 1st-10th floor. If there is a cost listed in A^2, then it will also be for sale in the Store on the Ground Floor and lower Stores as well. '01'–'04' for purchase in 4th Floor Store, etc. A '00' placed here means item is neither in Store nor can be found, but maybe assigned to a Player Class. E.g. the Mithril Mail in ***Halls***. Floors written in reverse notation, e.g. FF, FC, etc. = Wizard Class can use.

Sectors 0047 – 0048) *List of Magical Categories (Potions, Scrolls, Lanterns, etc.) from Quest*

Game Byte #	Bytes	Category Name	Bytes	Char Codes	Bytes	Floor*
00002614	94-9E	Touchstone	9F-A2	C8-C3-CA-CB	A3	01
00002624	A4-AE	Lantern	AF-B2	D2-C3-7B-7B	B3	01
00002634	B4-BE	Potion	BF-C2	7B-7B-D3-B6	C3	01
00002644	C4-CE	Scroll	CF-D2	B8-7B-B9-7B	D3	FF
00002654	D4-DE	Scroll	DF-E2	B8-7B-B9-7B	E3	FD
00002664	E4-EE	Wand	EF-F2	7B-B4-7B-B5	F3	04
00002674	F4-FE	Scroll	(0048) FF-02	B8-7B-B9-7B	03	FB
00002684	04-0E	Scroll	0F-12	B8-7B-B9-7B	13	F9

Up to 8 Categories of Magical Items may be defined. Category types, e.g. Scrolls, can be repeated in more than one of the 8 Categories if desired. Each Category has 5 Spells assigned to it (see the Table that follows). The Category bytes also provide the graphic representation (*character-code*) and the Floor on which the item first becomes (potentially) available. If the Floor number is in reverse notation, then only the Wizard (Ability) Class can use them. Example F9 = 7th Floor (when item can first be found), but only used by Characters with Wizard Abilities. *FF=1, FD=3, FB=5, F9=7, etc.

Sectors (0048 – 004A) List of 40 Magical Items from Quest

List #	Category	Game Byte #	Bytes	Name	Attributes	Spell	Uses	Max	Effect
1	**Touchstone**	00002694	**14-22**	Luck Stone	**23-25**	18	00	0F	Inc. Player Luck
2		000026A6	**26-34**	Honing Stone	**35-37**	0C	03	04	Inc. Weapon Damage
3		000026B8	**38-46**	Gem of Strength	**47-49**	00	00	04	Inc. Player HP
4		000026CA	**4A-58**	Omniscient Gem	**59-5B**	1C	00	14	Inc. Player Exp.
5		000026DC	**5C-6A**	Stumbling Rock	**6B-6D**	19	00	0A	Dec. Player Luck
6	**Lantern**	000026EE	**6E-7C**	Soothing Light	**7D-7F**	49	FA	FE	Dec. Party Healing Interval
7		00002700	**80-8E**	Aura of Warding	**8F-91**	41	FA	FF	Dec. Wandering Monster Prob.
8		00002712	**92-A0**	Rainbow Strobe	**A1-A3**	29	02	05	Dec. party Damage
9		00002724	**A4-B2**	Path Lightener	**B3-B5**	44	FA	FC	Inc. Ration Consumption Interval
A		00002736	**B6-C4**	Smudge Pot	**C5-C7**	40	FA	0A	Inc. Wandering Monster Prob.
B	**Potion**	00002748	**C8-D6**	Fortifying Brew	**D7-D9**	14	01	02	Inc. Weapon Bonus
C		0000275A	**DA-E8**	Reflex Draught	**E9-EB**	10	01	02	Inc. Player Armor Bonus
D		0000276C	**EC-FA**	Lightfoot Lager	**FB-FD**	18	01	04	Inc. Player Luck
E	(0049)	0000277E	**FE-0C**	Medicinal Ale	**0D-0F**	05	03	14	Dec. Player Damage
F		00002790	**10-1E**	Acid	**1F-21**	01	01	05	Dec. Player HP
10	**Scroll**	000027A2	**22-30**	Morale	**31-33**	4D	04	02	Dec. Monster Armor Protection
11		000027B4	**34-42**	Protection	**43-45**	51	04	02	Dec. Monster Attack Class
12		000027C6	**46-54**	ESP	**55-57**	74	04	00	Shows Monsters in Room
13		000027D8	**58-66**	Repair	**67-69**	08	03	FE	Inc. Player Armor Protection
14		000027EA	**6A-78**	Memory Absorber	**79-7B**	1D	01	0A	Dec. Player Exp.
15	**Scroll**	000027FC	**7C-8A**	Fire Ball	**8B-8D**	6D	06	08	Dec. All Monster HP
16		0000280E	**8E-9C**	Snare	**9D-9F**	61	08	03	Dec. Monster Mobility
17		00002820	**A0-AE**	Finds Traps	**AF-B1**	78	04	00	Finds Traps
18		00002832	**B2-C0**	Health Sheath	**C1-C3**	05	03	1E	Dec. Player Damage
19		00002844	**C4-D2**	Mad Scribble	**D3-D5**	21	01	02	Dec. Player Level
1A	**Wand**	00002856	**D6-E4**	Lightening Rod	**E5-E7**	71	06	0F	Dec. Monster HP
1B		00002868	**E8-F6**	Weapon Hoarder	**F7-F9**	34	08	00	Inc. Weapon Availability
1C	(004A)	0000287A	**FA-08**	Consuming Beam	**09-0B**	71	04	9C	Dec. monster HP
1D		0000288C	**0C-1A**	Staff of Life	**1B-1D**	29	02	14	Dec. Party Damage
1E		0000289E	**1E-2C**	Backfiring Beam	**2D-2F**	04	01	14	Inc. Player Damage
1F	**Scroll**	000028B0	**30-3E**	Lightening	**3F-41**	71	06	1E	Dec. Monster HP
20		000028C2	**42-50**	Haste	**51-53**	3C	04	01	Inc. Party Combat Speed
21		000028D4	**54-62**	Vulnerability	**63-65**	65	03	05	Dec. Monster Magical Resistance
22		000028E6	**66-74**	Goodwill	**75-77**	5C	03	04	Inc. Monster Bribability
23		000028F8	**78-86**	Despair	**87-89**	19	01	14	Dec. Player Luck
24	**Scroll**	0000290A	**8A-98**	Disintegrate	**99-9B**	6D	02	64	Dec. All Monster HP
25		0000291C	**9C-AA**	Nullify Power	**AB-AD**	59	04	32	Dec. Monster Special Power Chance
26		0000292E	**AE-BC**	Suppress Traps	**BD-BF**	79	06	00	Makes Traps Misfire
27		00002940	**C0-CE**	Restoration	**CF-D1**	05	04	64	Dec. Player Damage
28		00002952	**D2-E0**	Memory Drain	**E1-E3**	21	01	04	Dec. Player Level

Note: Each block of 5 Spells corresponds to one of the previous 8 Categories for Name (e.g. SCROLL) and Graphic representation.

Spell = Spell hex #'s are from list of the World's Known Spells**.**
Uses = maximum # of uses, or duration in paces. If '00' is used then the effect happens immediately upon contact.
Spells "# of uses" can be in Reverse Notation to specify an exact amount or duration.
Max = Maximum intensity or amount. This may refer to Damage, a Bonus, # of paces (as counter for duration – usually x Factor) depending on the nature of the Spell used. Some numbers are in Reverse Notation for an exact amount. E.g. FC=4, changes **002F**, byte 17 to a 6 for Consumption Interval. Changes in intervals are added to, or subtracted from, the current value.

Sector 004B - Quest Objects

Game	Quest	Item	Attribute		Game		Item	Attribute
Byte #	Bytes	Name	Bytes	Quest Attributes	Byte #	Bytes	Name	Bytes
00002996	16-20	King	21-28	02-0A-CC-C5-CD-C6-00-0C	000029E2	62-6C	Item #5	6D-74
000029A9	29-33	Rainbow Orb	34-3B	00-0A-CF-C7-BF-B7-00-0E	000029F5	75-7F	Item #6	80-87
000029BC	3C-46	Item #3	47-4E		00002A08	88-92	Item #7	93-9A
000029CF	4F-59	Item #4	5A-61		00002A1B	9B-A5	Item #8	A6-AD

Note: Quest Item names can be up to 11 characters long. **Attribute Bytes**: 1st = Floor where Item may 1st be found. 2nd Byte = Last Floor Item can be found. If the first and second Attribute Bytes are the same, then the Quest Item will only be found on that Floor. (If fewer Floors are selected for play than a specified Quest Item Floor, then the Item is placed on the lowest Floor available) 3rd through 6th Bytes = Char-Codes. Bytes = 7 & 8 = Time factor to find Item (Factor x # Floors selected = count down # to find a specific Quest object. E.g. 0C=120, 0E=140 if 10 floors are selected for play.

Sectors 004A-004B - Fountain Effect (Spells) and Probabilities from Quest

#	Game Byte #	Effect Byte	Fountain Effects (Sectors 004A - 004B)	Spell #	Bank 0 Bytes	Bank 0 % Prob.	Bank 1 Bytes	Bank 1 % Prob.	Bank 2 Bytes	Bank 2% Prob.	Bank 3 Bytes	Bank 3 % Prob.
1	00002964	**E4**	Decrease Player Damage	05	**EE**	28	**F8**	2D	**02**	32	**0C**	3C
2	00002965	**E5**	Increase Player Luck	18	**EF**	2C	**F9**	33	**03**	3A	**0D**	44
3	00002966	**E6**	Increase Player HP	00	**F0**	2E	**FA**	36	**04**	3E	**0E**	48
4	00002967	**E7**	Increase Player Experience	1C	**F1**	32	**FB**	3C	**05**	46	**0F**	50
5	00002968	**E8**	Increase Player Damage	04	**F2**	5A	**FC**	5A	**06**	5A	**10**	5A
6	00002969	**E9**	Decrease Player Luck	19	**F3**	5E	**FD**	5E	**07**	5F	**11**	5F
7	0000296A	**EA**	Decrease Player HP	01	**F4**	60	**FE**	60	**08**	5F	**12**	5F
8	0000296B	**EB**	Decrease Player Experience	1D	**F5**	64	**FF**	64	**09**	64	**13**	64
9	0000296C	**EC**			**F6**	(004B)	**00**		**0A**		**14**	
10	0000296D	**ED**			**F7**		**01**		**0B**		**15**	

Note: Up to 10 Fountain Effects (entered as hex digits), taken from the List of Spells may be entered. The Quest database only uses 8 effects leaving two blank (unused). There are four Banks of probabilities, numbered 0-3, that are used to determine the likelihood of a specific Fountain Effect occurring. In ***Quest***, Bank number 3 has the greatest probability of a beneficial effect occurring with Fountain use (hex 50=80%). Each successive Bank has a decreased probability of beneficial effects until Bank 0, which has a 50/50 chance of a beneficial effect (hex 32=50%). Regardless of which Bank a Fountain initially uses, over time and recurring use, the Bank of Probabilities being used drifts down to Bank 0.

Effects should be grouped into 'good' and 'bad' groups as seen above, with incremental probabilities occurring in each Bank. The last probability should always be Hex 64 or 100% for each Bank. Leave both the **Effect Byte** and the corresponding **probabilities** blank if you wish to have fewer than 10 Effects. Both Hallway and Room Fountains utilize this same List for Effects and Probabilities.

It is totally possible to mix or even reverse the effects as noted. For example, the first few effects listed could be definitely detrimental, but with repeated use, the Fountain would eventually generate "good effects" as the probabilities changed in favor of the last group of beneficial effects.

Sector 0044, Bytes 62-75 - *Initial Hallway Fountain Settings*

These 20 bytes (62-75) are used to select the initial Probability Bank (00-03) to be used by Hallway Fountains. There is room here for 20 Fountains, of which Quest assigns 2/ floor. Hallway Fountains in excess of 20 are assigned to Probability Bank 0.

Game Byte #

000022E2 Sector 0044 - Initial Hallway Fountain Settings from Quest

Byte	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F	70	71	72	73	74	75
Value	03	02	03	02	03	02	03	02	03	02	03	02	03	02	02	02	02	02	02	02

Individuals who are familiar with Quest will have noted that one of the two Hallway Fountains tends to be more beneficial (at least early on). The preceding table illustrates how this is done.

Room Fountains do not appear to have an initial Probability Bank option and seem to be randomly assigned to one by the program. In addition, they are numbered rather differently and this scheme is intermingled with that of "Living Statues". As previously noted any Bank assigned to a specific Fountain, over time, drifts down to Bank 0, or its equivalent.

Sector 004B - 004C List of Chest Traps from Quest

#	Game Byte #	Name Bytes	Trap Name	Effect Byte	Effect (Spell)	% Byte	% (Hex)	
1	00002A46	C6-CF	Explosion	D0	28	D1	0A	10%
2	00002A52	D2-DB	Dead Fall	DC	04	DD	1E	30%
3	00002A5E	DE-E7	Pit Trap	E8	04	E9	32	50%
4	00002A6A	EA-F3	Acid Spray	F4	09	F5	37	55%
5	00002A76	F6-FF	Fungus **Sector 004C**	00	01	01	41	65%
6	00002A82	02-0B	Forget Gas	0C	1D	0D	4B	75%
7	00002A8E	0E-17	Metal Mold	18	0D	19	50	80%
8	00002A9A	1A-23	Curse	24	15	25	55	85%
9	00002AA6	26-2F	Transmute	30	31	31	5F	95%
10	00002AB2	32-3B	Curse	3C	11	3D	64	100%

Note: Trap Names can be up to 10 characters in length. The **Effect Byte** contains the Spell number. The **% Byte** gives the probability (%) in hexadecimal that this effect will occur should the trap go off. The hexadecimal % should be in increasing probabilities and the last % should always be 64 in hex (100%). The final column gives the decimal % of the hex digits for convenience and ***is not*** part of the database. The severity of the Trap effect appears to be assigned by the module.

In the table above from **Quest** the ***Curse*** trap appears twice, but their effects when set off are different. The first ***Curse***, Spell 15 decreases Player Weapon Bonus and the second - Spell 11, decreases Player Armor Bonus.

Sectors 0047 – 0048) List of Magical (Tech) Categories from Sub12

Game Byte #	Name Bytes	Category Name	CharCode Bytes	Character-Codes	Floor Bytes	Floor
00002614	94-9E	RATION	9E-A2	B4 B5 B6 B7	A3	01
00002624	A4-AE	ADD-ON TECH	AF-B2	6E 7B 7B 98	B3	08
00002634	B4-BE	GRENADE	BF-C2	B8 7B 7B 7B	C3	08
00002644	C4-CE	L-II TECH	CF-D2	C5 7B 7B 7B	D3	01
00002654	D4-DE	MED KIT	DF-E2	D4 D5 D6 D7	E3	01
00002664	E4-EE	SMAW	EF-F2	E0 7B E1 7B	F3	F8
00002674	F4-FE	L-II WEAPON **Sector 0048**	FF-02	E2 7B E3 7B	03	F8
00002684	04-0E	R&D	0F-12	C0 C1 C2 C3	13	08

Highlighted Floors, written in reverse notation (F8=8th Floor), represent “reserved Class use” only Categories.

For a full explanation of *who can use what*, see the chapter on, **Creating Characters**.

X. Buying and Hoarding (a.k.a. Stores and Vaults)

Stores and Vaults will be considered as a unity, for this is how they are implemented within the ***Tunnels of Doom*** program and game databases. A maximum of five Stores ***or*** Vaults per Floor, or any combination of either, may be enlisted as part of your Dungeon design. (A Ground Floor Store is an exception and will be dealt with later.)

The desired quantity for both must be entered in **Sector 003A**, **Byte EE** (valid options, 00-05):

Game Byte #	Sector 003A Bytes	Default/ Controls
0000196E	**EE**	(03) Max # of Vaults & Stores per Floor (total of both)

However, certain conditions apply. Manipulation of this control byte simply reserves **"x"** number of rooms on each floor for Stores and Vaults. Once this choice is made then **"x"** number of rooms on each floor cannot be used for any other purpose, regardless if Stores or Vaults are actually placed there!

Using ***Quest*** as our example we find that ***Quest*** has **hex 03** as its value in this control byte. This sets a maximum of 3 Stores/ Vaults per floor. ***Quest*** has 2 Vaults located on each floor except for the 10th floor which has a third Vault. In addition, it has Stores on both the 4th and 8th Dungeon Levels (for a total of 03). Anyone who is familiar with ***Quest*** will recall the odd circumstance of a completely "empty room" on some of the floors. This is the trade-off that arises with "unused reservations". Once the number of rooms reserved for Stores and Vaults has been set, they must then be designated as active. Activation of Stores and Vaults is achieved by setting the appropriate bytes in **Sectors 0043-0044**:

Vault & Store Settings

Game Byte #	Sector 0043 Bytes		Game Byte #	Sector 0044 Bytes	
0000227E	**FE-02**	1st Floor	00002297	**17-1B**	6th Floor
00002283	**03-07**	2nd Floor (**Sector 0044**)	0000229C	**1C-20**	7th Floor
00002288	**08-0C**	3rd Floor	000022A1	**21-25**	8th Floor
0000228D	**0D-11**	4th Floor	000022A6	**26-2A**	9th Floor
00002292	**12-16**	5th Floor	000022AB	**2B-2F**	10th Floor

Note: Five Bytes are reserved for up to 5 Stores and/ or Vaults per Floor. If a Byte contains: 01 = Store, 02 = Vault. To be functional the corresponding Vault Combinations will need to be set, and Byte '**EE**' in **Sector 003A** also needs to be set correctly.

If a Store or Stores are desired, set these first with hex 01. If only vaults then set as hex 02. If neither is desired leave the corresponding byte 00. To complete the process the corresponding byte in **Sector 0044** must be set with the Vault's combination parameters. There is one byte used per Vault for the combination. The first hex digit of the byte sets the length or number of digits in the combination. The second hex digit of the byte sets the combination range, from 1 to n. Where n is a number from 1 to 9. See example below.

Vault Combination Settings

Game Byte #	Sector 0044 Bytes		Game Byte #	Sector 0044 Bytes	
000022B0	**30-34**	1st Floor	000022C9	**49-4D**	6th Floor
000022B5	**35-39**	2nd Floor	000022CE	**4E-52**	7th Floor
000022BA	**3A-3E**	3rd Floor	000022D3	**53-57**	8th Floor
000022BF	**3F-43**	4th Floor	000022D8	**58-5C**	9th Floor
000022C4	**44-48**	5th Floor	000022DD	**5D-61**	10th Floor

Note: Five Bytes are reserved per floor (for up to 5 Vaults). Bytes correspond to specific Vaults specified above. E.g. Byte 30 sets the Combination range for the Vault in Byte 'FE' of Sector 0043, etc. For each Byte: 1st digit = length (#) of combination, 2nd digit = range of numbers. For example a '34' = Combination of 3 numbers, ranging from 1- 4. The range always assumes it begins with one. If a Store '01' is present, then leave this field 00.

Returning to the ***Quest*** implementation, you may recall that floors 4 and 8 did not have an empty room, due to the presence of a Store. Floor 10 does not as well, due to the presence of a 3rd Vault - but neither did the first or second floors? The solution for floors 1 and 2 was found to be the odd occurrence of a `hex 01` residing in the location where the first Vault combination should have been, however there was no corresponding Vault located there. This was likely caused by the database programmer attempting to create a Store with `hex 01`, but placed this digit in the Vault combination bytes location instead of where the Store/ Vault activation bytes are kept. And so an unforeseen, new feature to TOD was made…

There will very likely be instances when you do not care to place the full complement of Stores or Vaults on every floor, especially when the control byte is set to 4 or 5. But you may also wish to avoid "empty rooms". Fortunately, there is a work around for this. These potentially empty rooms can be seeded with objects from your lists of Weapons, Armor and Magical Items. These seeds are entered into the bytes otherwise reserved for Vault combinations (and so cannot be used to seed a "Store"). Use the following tabulation as guide:

Hex		Places:
01	=	Magical Item
03	=	2 Magical Items
04	=	Armor
05	=	Magical Item & Armor
07	=	2 Magical Items & Armor
08	=	Weapon
09	=	Magical Item & weapon
0B	=	2 Magical Items & Weapon
0C	=	Weapon & Armor
0D	=	Magical Item, Armor & Weapon

An explanation of ***why*** these hex bytes will produce the results indicated is somewhat complex, and will be deferred to the chapter on *Creating Scenarios* (to place it in proper context). It should be stressed that it is *very* unlikely that the creator of the ***Tunnels of Doom*** program ever intended this as a design "option". Hence, use at your own risk. However, I have never noted any ill effects in game creation/ execution using this technique. There is also a method to "force" the appearance of a Chest in these otherwise empty rooms, in which case your seed items will appear therein. This will be explained in the chapter on *Dungeon Design*.

A few closing words on Stores and Vaults

It is probably not so-common knowledge that any room designated to be a Store will actually only produce one if the Party has some gold or currency when entering that room. If they do not, they are treated to the Game Logo graphic upon entering, but that's it! The graphic representations for Stores (a.k.a. Game Logo) and Vaults are part of Graphics Bank #2, **Sector 004D**.

Game	Sector 004D - *Graphic Character Space for Room Contents*			
Byte #	Bytes			
00002B90	**10-2F**	Char-Codes 90-93	Game Logo Graphic, Color set by Byte 82	(01)
00002BB0	**30-4F**	Char-Codes 94-97	Vault Graphics, Color set by Byte 82	(02)

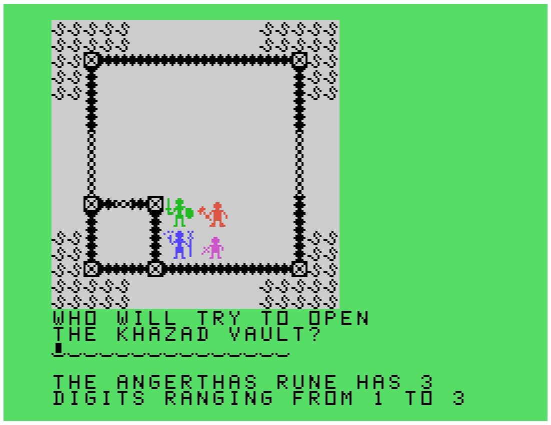
The hex number in parentheses, following the color control byte, is how the TOD program references the graphic for a Store or Vault. Each hex number accesses the group of 4 *character-codes* that represent the graphic. This information is provided as explanation for why hex 01 = Store, 02 = Vault, etc. in the database. This direct correspondence is also the reason the memory locations for Game Logo (Store) and Vaults cannot be relocated. Plotting successive hex numbers (in the area designated for Stores and Vaults) will continue to call forth successive groups of 4 *character-codes*, but will also frequently lock up the system! (If you just gotta do it – go ahead!)

Stores and Vaults - Keywords

Game	Sector 0052		
Byte #	Bytes	Word	
0000312C	AC-B7	VAULT	12 Bytes

Modifying the word VAULT. Used in the phrase:
“WHO WILL TRY TO OPEN THE . . . ?”

Game	Sector 002F
Byte #	Bytes
00000DE2	62-71 Word "Combination" - 16 bytes



WHO WILL TRY TO OPEN
THE KHAZAD VAULT?

THE ANGERTHAS RUNE HAS 3
DIGITS RANGING FROM 1 TO 3

Examples of modifications from Halls

Modifying the word COMBINATION in Sector 002F, which forms part of the phrase:
“THE COMBINATION HAS 3 DIGITS RANGING FROM 1 TO 3”

Also located in **Sector 002F**, in the byte immediately preceding the above, is the probability factor, in hex, for not receiving any wounds when you guess incorrectly at Vault combinations:

Byte #	Byte
00000DE1	61 (28) % Probability of not receiving wounds at Vaults.

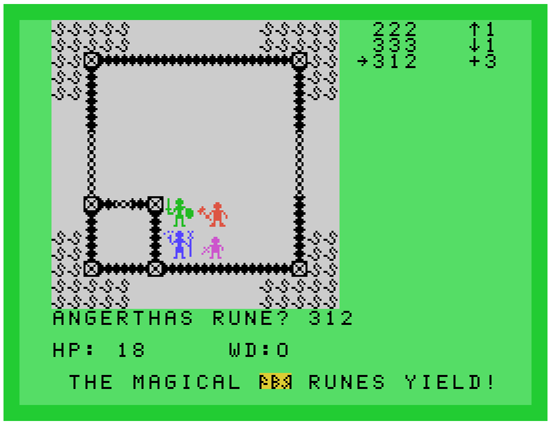
The default probability (from ***Quest***) is `hex 28`, or 40%

If wounds are received the amount is calculated as follows:

Game Difficulty	Wounds
Easiest	1 wound for every incorrect digit in your guess
Medium	2 wounds for every incorrect digit in your guess
Hardest	3 wounds for every incorrect digit in your guess

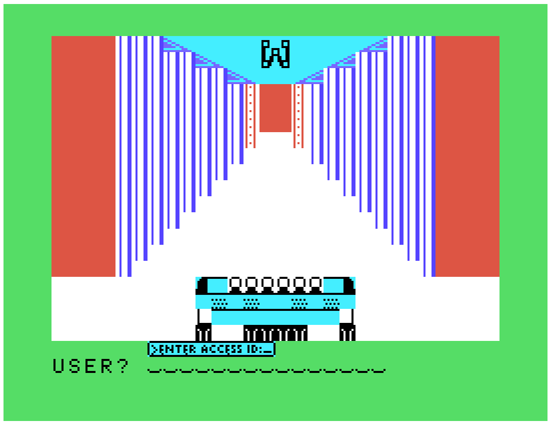
It is probably not common knowledge that both Hero and Rogue Classes have increased probabilities of wound avoidance with incorrect combination guesses. This is alluded to, only once, in the TOD game manual on page 20 where it states the Rogue, “Has a better chance of avoiding traps on vaults and chests.” In the paragraph immediately below it adds that the Hero Class, “has the Rogue’s trap avoidance ability”. For both these Classes the number of wounds received is not decreased, but the likelihood that you will not receive any at all is increased. (If all else remains equal, this can be upwards of 1/3 of the incorrect guesses.)

A final word on Stores, Vaults and virtually anything else that generates modifiable dialogue – simple graphics can usually be imbedded within the dialogue itself. The primary requirement being that the *pattern-identifiers*, which define the graphic must be placed in the correct graphics bank for access. The appropriate corresponding *character-codes* must also be used. For examples of using this technique, see both ***Halls*** and ***Sub12*** databases.

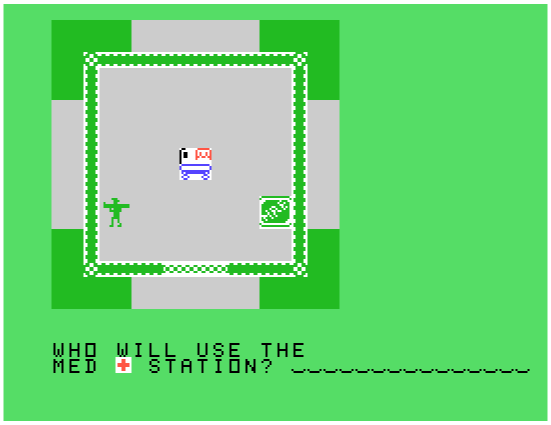


```
                        222    ↑1
                        333    ↓1
                       →312    +3
ANGERTHAS RUNE? 312
HP: 18      WD:0
 THE MAGICAL    RUNES YIELD!
```

Examples of Imbedding Graphics within a Dialogue



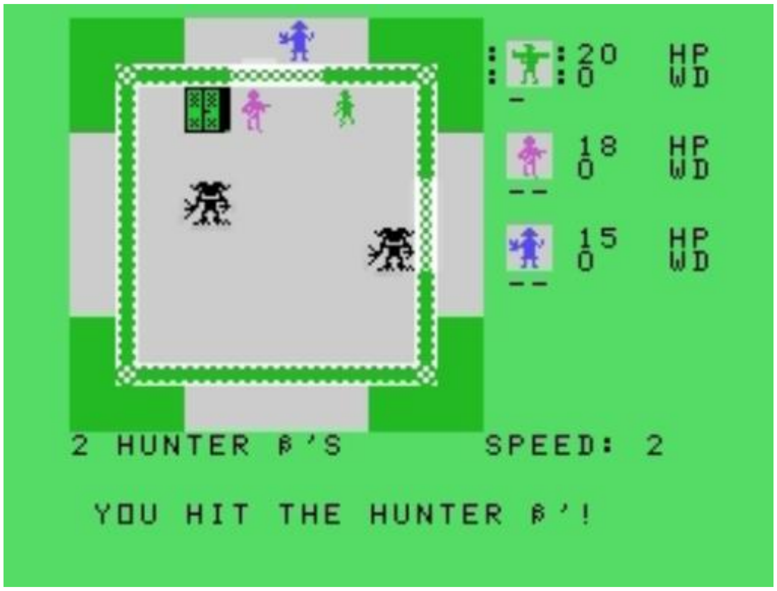
```
      >ENTER ACCESS ID:_
USER? _______________
```



```
WHO WILL USE THE
MED ✚ STATION? _______________
```

Imbedding a Graphic within a Dialogue

Imbedding a Monster's Name



```
          :20   HP
          :0    WD

           18   HP
           0    WD

           15   HP
           0    WD

2 HUNTER β'S        SPEED: 2

 YOU HIT THE HUNTER β'!
```

Graphic within a (Greek Beta symbol)

XI. 56 Reasons to Avoid Dungeons

Monster Distribution and its Significance in the Quest Database (Important)

#	Experience	Name	Lev-1	Lev-2	Lev-3	Lev-4	Lev-5	Lev-6	Lev-7	Lev-8	Lev-9	Lev-10	Lev-11	Lev-12
1	10	Goblin	1											
2		Kobold	■											
3		Rat	■											
4		Ooze	■											
5	30	Evil Mane	■	2										
6		Giant Rat	■	■										
7		Lizard	■	■										
8		Imp	■	■										
9	60	Orc	■	■	3									
10		Skeleton	■	■	■									
11		Wild Dog	Floor	■	■									
12		Spider	■	■	■									
13	100	Wolf	■	■	■	4								
14		Zombie	■	■	■	■								
15		Dark Slime	■	Floor	■	■								
16		Spider	■	■	■	■								
17	150	Lemure	■	■	■	■	5							
18		Wight	■	■	■	■	■							
19		Dust Devil	■	■	Floor	■	■							
20		Gremlin	1	■	■	■	■							
21	210	Cursed One		■	■	■	■	6						
22		Metalloid		■	■	■	■	■						
23		Snake		■	■	Floor	■	■						
24		Vampire Bat		2	■	■	■	■						
25	280	Smogg			■	■	■	■	7					
26		Pixie			■	■	■	■	■					
27		Hobgoblin			■	■	Floor	■	■					
28		Ghost			3	■	■	■	■					
29	360	Metazoid				■	■	■	■	8				
30		Land Crab				■	■	■	■	■				
31		Whiplash				■	■	Floor	■	■				
32		Gnoll				4	■	■	■	■				
33	450	Troll					■	■	■	■	9			
34		Shambler					■	■	■	■	■			
35		Serpent					■	■	Floor	■	■			
36		Minotaur					5	■	■	■	■			
37	550	Ghoul						■	■	■	■	10		
38		Giant Wasp						■	■	■	■	■		
39		Ogre						■	■	Floor	■	■		
40		Devourer						6	■	■	■	■		
41	660	Vampire							■	■	■	■		
42		Scorpion							■	■	■	■		
43		Buzz Bomb							■	■	Floor	■		
44		Hill Giant							7	■	■	■		
45	780	Frost Giant								■	■	■		
46		Demon								■	■	■		
47		Tarantula								■	■	Floor		
48		Wyvern								8	■	■		
49	810	Demon King									■	■		
50		Dragon									■	■		
51		Elemental									■	■		
52		Pit Fiend									9	■		
53	950	Arch-Devil										■		
54		Land Shark										■		
55		Dragon Lord										■		
56		Will O'Wisp										10		

XI. 56 Reasons to Avoid Dungeons

From the perspective of my own Tunnels of Doom experience, creating the TOD monsters and their stats was a defining moment. The original Editor that I received from Asgard Software only permitted modification of "51 Monster" profiles... This caused issues in the execution of 10-level Dungeon games in that "ghosts from the past and out of context" invariably appeared. A later update, which I unfortunately never heard about (at the relevant time), increased the tally to 55.

Examining the ***Quest*** game with a Sector Editor revealed that 56 monster types potentially needed to be accounted for in any ***Quest*** style dungeon scenario... this led to further research. As it turned out the number 56 is not capricious, but is based on the Table of data, which opened this chapter.

In examining the Table you will quickly note the following:

- Monsters are grouped by “Experience”. This represents the Experience Points a Player receives by successfully vanquishing a particular Monster. There are four Monsters assigned to each Experience level.
- It is presumed that the greater the Experience level assigned to a Monster, the more difficult they will be to deal with… however, this rule is neither absolute nor always intelligently applied. The reason being that Experience level is assigned by a Monster’s placement in a List, while actual Monster toughness and difficulty is based upon their individually assigned stats.
- Five Experience levels of Monsters are assigned to each floor.
- On every Floor the Monsters are selected from a pool of 20 Monster Types.
- Progression to each Floor involves the dropping of the lowest Experience level of Monsters from the pool and adding the next highest Experience level to the pool.
- If this step-wise sequence is continued for a 10 Floor Dungeon – 56 Monster Types will be necessary!
- By using this Table you can quickly determine how many Monster Types you will need to create to populate your Dungeon, based upon the maximum number of Floors you set. E.g. if you wished to populate the ***Pennies & Prizes*** Database with Monsters you need not define more than 32, for any entered beyond that number would never be used in a 4 Floor Dungeon.
- The number of Rooms/ Floor does not influence this sequence.

The variety of “Monsters” and their capabilities is another intriguing aspect of each Tunnels of Doom game database. The efficient use of both Module and VDP memory makes this possible, with the latter adding significantly to game flexibility. In addition to a Name, 14 specific characteristics may be assigned to each Monster type and several of them may randomly or intentionally be modified during game play. We shall proceed to an examination of these characteristics, which begin in **Sector 0030, byte CA** (on Disk) or with **Game Byte #00000F4A** using the first entry as an example of how all are considered.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA-D5	D6	D7	D8	D9	DA	DB	DC	DD^1	DD^2	DE^1	DE^2	DF^1	DF^2

1) Monster Name:

- The Monsters are not numbered in the database. This is provided as a matter of convenience. Monster Names can be up to 12 characters. If >1 is present, the name is automatically pluralized.
- Simple Graphics can be imbedded within the name. (See **Chapter X** and ***Sub-12*** for examples)
- Monster Names can be used more than once within the List, with the same or different stats and characteristics.

We will now consider the 10 bytes defining the 14 Characteristics or Attributes that are (potentially) assigned to each Monster category.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA-D5	D6	D7	D8	D9	DA	DB	DC	DD[1]	DD[2]	DE[1]	DE[2]	DF[1]	DF[2]

2) Level:

- ➢ This represents the Monster equivalent of Hit Points. When Monsters are placed into the Combat Queue they are each (and only then) assigned individual HP based upon HP= Level*(RND*6) with the maximum Hit Points always being (Level*6) for any Monster. This provides necessary variation in game play and challenge.
- ➢ When a Monster incurs wounds these are subtracted from its Hit Points. When the Hit Points level drops to zero, that monster is dead. (Yes, some are easier to slay than others!)
- ➢ A Monster with a Level of 01 will have a maximum of 6 Hit Points. One with a level of hex 0F may potentially have a maximum of 90 Hit Points or hex 5A. Be sensible in Monster Level progression.
- ➢ A Monster's Level is always expressed as "Maximal Hit Point" in Monster Reports.
- ➢ Spells hex 6C, 6D, 70, 71 directly affect the Hit Points of Monsters in the Combat Queue.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA-D5	D6	D7	D8	D9	DA	DB	DC	DD[1]	DD[2]	DE[1]	DE[2]	DF[1]	DF[2]

3) Defense Value:

- ➢ Defense Value is the Monster equivalent of Player Armor Protection. The higher the DV, the more likely you will miss your intended target.
- ➢ The DV does not minimize the damage received, but the likelihood of a successful Player attack.
- ➢ A Monster's DV is expressed as "Defense" in Monster Reports.
- ➢ Spells hex 4C & 4D "Monster Armor Protection" affect their Defense Value.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA-D5	D6	D7	D8	D9	DA	DB	DC	DD[1]	DD[2]	DE[1]	DE[2]	DF[1]	DF[2]

4) Attack Value:

- ➢ While a Monster's DV factors into the probability of its being hit, the AV is a measure of its ability to successfully strike a Player.
- ➢ While a Player's unsuccessful attack is graphically apparent during game play, a Monsters unsuccessful attack is demonstrated by a "flicker" of the Monster graphic (like it intended to move, but did not).
- ➢ An Attack Value entered in Reverse Notation enables a Monster to attack a Player from anywhere within the room. I.e. a Ranged Attack.
- ➢ It goes without saying that providing Monsters with both a high DV and AV too early in a game will likely result in player frustration!
- ➢ A Monster's AV is expressed as "Attack" in Monster Reports.
- ➢ Spells hex 50 & 51 "Monster Attack Class" affects their Attack Value.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA–D5	D6	D7	D8	D9	DA	DB	DC	DD^1	DD^2	DE^1	DE^2	DF^1	DF^2

5) Damage:

- ➢ Represents the <u>maximum</u> amount of damage that can be inflicted when successfully targeting a Player. This is equivalent to a Player's combined Weapon and Weapon Bonus values.
- ➢ The rating of maximum Damage amount does not factor into successful targeting.
- ➢ A Monster's Damage is expressed as "Maximum Attack Damage" in <u>Monster Reports</u>.
- ➢ Spells hex 54 & 55 "Monster Attack Damage" affects their Damage Value.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA–D5	D6	D7	D8	D9	DA	DB	DC	DD^1	DD^2	DE^1	DE^2	DF^1	DF^2

6) Special Attack %:

- ➢ The probability or chance, expressed in hexadecimal, that a Monster will use its Special Attack ability. E.g. 32 (hex) = 50% probability.
- ➢ Monster Special Attacks are an excellent means for taking a gamer unawares, for their special ability is unknown unless the SA is used, or the gamer presses "3" for a Monster Status Report.
- ➢ If a Special Attack ability is undesired, leave the SA%, SA Type and SA Damage bytes 00.
- ➢ A Monster's SA% is expressed as "Special Attack Chance" in <u>Monster Reports</u>.
- ➢ Spells hex 58 & 59 "Monster Attack Class" affects their "Special Power Chance" or SA%.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA–D5	D6	D7	D8	D9	DA	DB	DC	DD^1	DD^2	DE^1	DE^2	DF^1	DF^2

7) Special Attack Type:

- ➢ Special Attack Type is a hex value from 01 to 14 that selects the <u>corresponding</u> Special Attack from the List of Monster Special Attacks. These hex numbers are not part of the List, but inferred from their ordinal position (memory location).
- ➢ The Spell #'s in the List are derived from the: **List of the World's Known Spells . . . recognized by TOD**
- ➢ If Special Attacks are not used, leave this byte 00.
- ➢ A Special Attack Type (its hex #), written in Reverse Notation, permits a Ranged Special Attack.

Monster Special Attack List from ***Quest***

#	Game Byte #	Name Bytes	Spell Name	Spell Bytes	Spell #	Effect
1	0000141A	Sector 0035 9A–A8	Tongue of Flame	A9	04	Inc. Player Damage
2	0000142A	AA–B8	Curse	B9	19	Dec. Luck
3	0000143A	BA–C8	Phase Shift	C9	4C	Inc. Monster Armor Protection
4	0000144A	CA–D8	Rage	D9	50	Inc. Monster Attack Class
5	0000145A	DA–E8	Corrosion	E9	09	Dec. Player Armor Protection
6	0000146A	EA–F8	Sand Blast	F9	09	Dec. Player Armor Protection
7	0000147A	FA–08	Poison Bite	09	04	Inc. Player Damage
8	0000148A	0A–18	Poison Sting **(Sector 0036)**	19	04	Inc. Player Damage
9	0000149A	1A–28	Metal Eating	29	0D	Dec. Player Weapon Damage
A	000014AA	2A–38	Blood Drain	39	01	Dec. Player HP
B	000014BA	3A–48	Suffocation	49	04	Inc. Player Damage

C	000014CA	4A-58	Prank	59	19	Dec. Luck
D	000014DA	5A-68	Life Drain	69	1D	Dec. Player experience
E	000014EA	6A-78	Breathes Fire	79	04	Inc. Player Damage
F	000014FA	7A-88	Regeneration	89	70	Inc. Monster HP
10	0000150A	8A-98	Earth Quake	99	28	Inc. Party Damage
11	0000151A	9A-A8	Crushing Chomp	A9	04	Inc. Player Damage
12	0000152A	AA-B8	Pitchfork	B9	01	Dec. Player HP
13	0000153A	BA-C8	Tail Whip	C9	04	Inc. Player Damage
14	0000154A	CA-D8	Constriction	D9	04	Inc. Player Damage

The Special Attack # is inferred from its ordinal position within the List. It is not an element of the List.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA-D5	D6	D7	D8	D9	DA	DB	DC	DD^1	DD^2	DE^1	DE^2	DF^1	DF^2

8) Special Attack Damage:

- The Special Attack Damage represents the "amount" or "duration" (in paces/ key presses) of the Special Attack Type. (See Chapter II for details.)
- Using a Special Attack that references a Spell's *amount* or *quantity* in "*duration*", e.g. "Party Combat Speed" should be made with extreme care. It is often better to use Reverse Notation in these instances to impose a specific, but limited duration. In our example hex FF (for 1 x 10 or ten paces) or FE (for 2 x 10 or twenty paces) would be appropriate. If this is coupled with a low SA%, so that a Party is not repeatedly zapped with the same Special Attack, all should be well.
- It should be noted that ***Quest*** (rather wisely) shied away from Spells which utilized a duration factor in their Monster Special Attacks.

The next six or seven Monster Characteristics are unique in that the individual hex digits of each byte, are used to express distinct characteristics.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA-D5	D6	D7	D8	D9	DA	DB	DC	DD^1	DD^2	DE^1	DE^2	DF^1	DF^2

9) Monster Graphic:

- The eighth attribute byte of each Monster type defines both the Graphic and associated Sound to be employed when this Monster is in the Combat Queue. The first hex digit defines the Graphic and the second hex digit the Sound.
- The Monster Graphic is defined by hex digits 0-F and these 16 digits refer to the Defense and Attack Graphic poses defined in **Sectors 0036-003A**. Again, the hex digits 0-F define their ordinal location in VDP memory and are not part of the List itself.
- The Monster Attack Graphic is shown whenever a successful attack is made to one or more Party members and whenever a successful attack is made by a player to a Monster (flickers briefly).
- In the Monster Combat Queue the order of the hex digits for graphics and sound exchange places, presumably because you can "hear" a Monster, by pressing "L" before you actually see them.
- Monster graphics are paged into a temporary workspace area, represented on disk by **Sector 004D**, bytes 80-DF, for their interaction with Party members.

Monster Graphics

(Monster Defense/ Attack Pose Graphics. The Graphic hex # found in the list of 56 Monsters refers to this List)

	Game	Sector 0036			Game	Sector 0038	
#	Byte #	Bytes	D/ A Pose		Byte #	Bytes	D/ A Pose
0	0000155A	DA-F9	Defense	8	0000175A	DA-F9	Defense
	0000157A	FA-19	A Sector 0037		0000177A	FA-19	A Sector 0039
1	0000159A	1A-39	Defense	9	0000179A	1A-39	Defense
	000015BA	3A-59	A		000017BA	3A-59	A
2	000015DA	5A-79	Defense	A	000017DA	5A-79	Defense
	000015FA	7A-99	A		000017FA	7A-99	A
3	0000161A	9A-B9	Defense	B	0000181A	9A-B9	Defense
	0000163A	BA-D9	A		0000183A	BA-D9	A
4	0000165A	DA-F9	Defense	C	0000185A	DA-F9	Defense
	0000167A	FA-19	A Sector 0038		0000187A	FA-19	A Sector 003A
5	0000169A	1A-39	Defense	D	0000189A	1A-39	Defense
	000016BA	3A-59	A		000018BA	3A-59	A
6	000016DA	5A-79	Defense	E	000018DA	5A-79	Defense
	000016FA	7A-99	A		000018FA	7A-99	A
7	0000171A	9A-B9	Defense	F	0000191A	9A-B9	Defense
	0000173A	BA-D9	A		0000193A	BA-D9	A

As there are 16 Monster graphic representations available and 56 potential Monster types, it is obvious that some will have to be used more than once! A useful approach when creating Monster graphics is to keep in mind representations that can easily be applied to more than one Monster type. For instance, the same graphic could easily be used to represent a Mad Dog, Wolf, or Hyena, but not so much a Scorpion.

A Wizard, Sorcerer, Magician or Mage can easily adapt to the same representation. So would a Sprite, Imp, Hobgoblin or Pixie. In short, create adaptable representations that are applicable to multiple villains. This will help maintain scenario interest.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA-D5	D6	D7	D8	D9	DA	DB	DC	DD^{1}	DD^{2}	DE^{1}	DE^{2}	DF^{1}	DF^{2}

10) Monster Sound:

- The Tunnels of Doom program puts the TI99/4a's sound capabilities to good use; both with game play music and sound effects.
- The <u>second</u> hex digit of the eighth attribute byte refers to a Sound Table located in **Sector 0043**, these in turn elicit the appropriate sound effect from the TOD program:

```
Game       Sector 0043
Byte #     Bytes            A4                                    ASCII
00002224  A0  52 45 20 20 00 01 02 03 04 05 06 07 08 09 0A 0B  RE  ............
00002230  B0  0C 0D 0E 0F C7 F2 00 00 00 00 00 00 00 00 00 00  ................
                         B3
```

00002224 **A4-B3** Pointers to the 16 Monster sounds available in the TOD Module. The Sound hex digit found in the list of 56 Monsters refers to this List, which in turn cues the TOD Module for the desired sound effect. ***Default:*** hex 00 to 0F.

Again, organizing your Monster sounds to correlate with specific graphics requires a planned approach.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA-D5	D6	D7	D8	D9	DA	DB	DC	DD^{1}	DD^{2}	DE^{1}	DE^{2}	DF^{1}	DF^{2}

11 & 12) Monster Mobility & Negotiation Probability:

- Monster Mobility and Negotiation % are the shared product of the first hex digit of the ninth Monster attribute byte.
- Mobility is the probability that a Monster will move towards you to attack if you are not adjacent to them and they do not posses ranged attack abilities.
- Negotiation is the probability that a Monster can be bribed to let you pass.
- Baseline values for these characteristics are: 25, 50, 75 and 100% for Mobility. 0, 25, 50 and 75% for Negotiation. These are assigned through placement of the appropriate hex digit:

First	Results:	
Hex Digit	Mobility	Negotiation
0	25%	0%
1	50%	0%
2	75%	0%
3	100%	0%
4	25%	25%
5	50%	25%
6	75%	25%
7	100%	25%
8	25%	50%
9	50%	50%
A	75%	50%
B	100%	50%
C	25%	75%
D	50%	75%
E	75%	75%
F	100%	75%

- A Monster's "Mobility" and "Negotiation" are expressed as such in Monster Reports.
- Spells hex 5C & 5D "Monster Bribability" affects a Monster's "Negotiation" chance.
- Spells hex 60 & 61 "Monster Mobility" affects a Monster's "Mobility" chance.

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA-D5	D6	D7	D8	D9	DA	DB	DC	DD^{1}	DD^{2}	DE^{1}	DE^{2}	DF^{1}	DF^{2}

13) Monster Speed:

- A Monster's Speed is regulated in the same fashion as the Players/ Characters with the exception that any change in speed persists for the current combat/ encounter only, whereas the duration for Players is measured in key presses.
- Speed represents the number of "actions a Monster can take in a round of combat. The number of attacks… is always one less than this number."
- The same precautions expressed for "Special Attack Damage" are applicable here.
- A Monster's "Speed" is expressed as Speed in Monster Reports.
- Spells hex 68 & 69 "Monster Combat Speed" affects a Monster's "Speed".

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic	Sound	Mob / Neg	Mag Res	Speed	?
1	00000F4A	Sector CA-D5	D6	D7	D8	D9	DA	DB	DC	DD^{1}	DD^{2}	DE^{1}	DE^{2}	DF^{1}	DF^{2}

14) ???:

- The 14th and last Monster attribute, which is defined in the second hex digit of its byte is somewhat of a mystery, inasmuch as nothing seems to has been *left undefined* in terms of Monster attributes as determined by the Monster Status Report.
- Quest provides hex digits, in a generally ascending order, from 0-A for this characteristic. If these digits express probabilities, from 0 to 100%, one would expect their distribution to be more random instead of a generally ascending order.
- If the last hex digit represents an absolute value, from 0-10, then this must represent the amount or quantity of an attribute, but none of the obvious ones remain unaccounted.
- Whatever this value represents, it is passed along with the 13 others to the Monster Combat Queue, but here its order is reversed with that of the hex digit for Speed.
- There is a possibility that this hex number represents the counterpart of a Player's (Increased) "Luck" attribute, but applied to a ***type*** of Monster instead of individual Monster characters. For Player Characters, Luck increases and decreases from an assigned baseline, but the "amount" of increase or decrease or the total quantity is never expressly stated. It is reasonable to assume that the most dangerous Monsters, the lowest in the List, might posses the greatest Luck?
- If this value does indeed represent an increase in Monster "Luck", then altering its counterpart in the Monster Combat Queue should change the Monster verses Character combat advantage %, but unfortunately this does not appear to happen.
- An alternate explanation is that the original intent for the database was to use separate hex digits for the Mobility and Negotiation attributes, with Negotiation (or both Mobility & Negotiation) being from 0-100% (0-A in hex). Due to challenges with the game (module) implementation of the database this was later altered to 4 selections (%) for both Mobility and Negotiation. This should have simplified the programming. In the Monster Status Report "Negotiation" follows "Speed", which would seem to indicate this byte was intended to provide information on the latter.
- For every Player attribute, including Luck, there is a Spell that can raise or lower its value. The same holds true with Monster attributes. However, there is no Spell that will Increase or Decrease a Monster's Luck. Can it even be done apart from an initial baseline value for all Monsters?
- There do not appear to be any Spells, affecting Monsters, which remain unaccounted for when matched to their attributes!
- In short, though there are hex values there, it is unclear if they actually do anything…
- More investigation is clearly necessary…

The first table that follows provides the byte locations for all the attributes of the 56 Monster types.

The second table, from ***Quest***, can be used as a guide rule for assigning your own Monster attributes. The last and unknown attribute hexes are highlighted when assigned to a Monster type:

Characteristics of 56 Monster Types - Byte References

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic 1st hex	Sound 2nd hex	Mob / Neg -1st	Mag Res -2nd	Speed	?
1	00000F4A	CA-D5	D6	D7	D8	D9	DA	DB	DC	DD^1	DD^2	DE^1	DE^2	DF^1	DF^2
2	00000F60	E0-EB	EC	ED	EE	EF	F0	F1	F2	F3	F3	F4	F4	F5	F5
3	00000F76	F6-01	02	03	04	05	06	07	08	09	09	0A	0A	0B	0B
4	00000F8C	Sec 0031 0C-17	18	19	1A	1B	1C	1D	1E	1F	1F	20	20	21	21
5	00000FA2	22-2D	2E	2F	30	31	32	33	34	35	35	36	36	37	37
6	00000FB8	38-43	44	45	46	47	48	49	4A	4B	4B	4C	4C	4D	4D
7	00000FCE	4E-59	5A	5B	5C	5D	5E	5F	60	61	61	62	62	63	63
8	00000FE4	64-6F	70	71	72	73	74	75	76	77	77	78	78	79	79
9	00000FFA	7A-85	86	87	88	89	8A	8B	8C	8D	8D	8E	8E	8F	8F
10	00001010	90-9B	9C	9D	9E	9F	A0	A1	A2	A3	A3	A4	A4	A5	A5
11	00001026	A6-B1	B2	B3	B4	B5	B6	B7	B8	B9	B9	BA	BA	BB	BB
12	0000103C	BC-C7	C8	C9	CA	CB	CC	CD	CE	CF	CF	D0	D0	D1	D1
13	00001052	D2-DD	DE	DF	E0	E1	E2	E3	E4	E5	E5	E6	E6	E7	E7
14	00001068	E8-F3	F4	F5	F6	F7	F8	F9	FA	FB	FB	FC	FC	FD	FD
15	0000107E	Sec 0032 FE-09	0A	0B	0C	0D	0E	0F	10	11	11	12	12	13	13
16	00001094	14-1F	20	21	22	23	24	25	26	27	27	28	28	29	29
17	000010AA	2A-35	36	37	38	39	3A	3B	3C	3D	3D	3E	3E	3F	3F
18	000010C0	40-4B	4C	4D	4E	4F	50	51	52	53	53	54	54	55	55
19	000010D6	56-61	62	63	64	65	66	67	68	69	69	6A	6A	6B	6B
20	000010EC	6C-77	78	79	7A	7B	7C	7D	7E	7F	7F	80	80	81	81
21	00001102	82-8D	8E	8F	90	91	92	93	94	95	95	96	96	97	97
22	00001118	98-A3	A4	A5	A6	A7	A8	A9	AA	AB	AB	AC	AC	AD	AD
23	0000112E	AE-B9	BA	BB	BC	BD	BE	BF	C0	C1	C1	C2	C2	C3	C3
24	00001144	C4-CF	D0	D1	D2	D3	D4	D5	D6	D7	D7	D8	D8	D9	D9
25	0000115A	DA-E5	E6	E7	E8	E9	EA	EB	EC	ED	ED	EE	EE	EF	EF
26	00001170	F0-FB	FC	FD	FE	FF	00	01	02	03	03	04	04	05	05
27	00001186	Sec 0033 06-11	12	13	14	15	16	17	18	19	19	1A	1A	1B	1B
28	0000119C	1C-27	28	29	2A	2B	2C	2D	2E	2F	2F	30	30	31	31
29	000011B2	32-3D	3E	3F	40	41	42	43	44	45	45	46	46	47	47
30	000011C8	48-53	54	55	56	57	58	59	5A	5B	5B	5C	5C	5D	5D
31	000011DE	5E-69	6A	6B	6C	6D	6E	6F	70	71	71	72	72	73	73
32	000011F4	74-7F	80	81	82	83	84	85	86	87	87	88	88	89	89
33	0000120A	8A-95	96	97	98	99	9A	9B	9C	9D	9D	9E	9E	9F	9F
34	00001220	A0-AB	AC	AD	AE	AF	B0	B1	B2	B3	B3	B4	B4	B5	B5
35	00001236	B6-C1	C2	C3	C4	C5	C6	C7	C8	C9	C9	CA	CA	CB	CB
36	0000124C	CC-D7	D8	D9	DA	DB	DC	DD	DE	DF	DF	E0	E0	E1	E1
37	00001262	E2-ED	EE	EF	F0	F1	F2	F3	F4	F5	F5	F6	F6	F7	F7
38	00001278	Sec 0034 F8-03	04	05	06	07	08	09	0A	0B	0B	0C	0C	0D	0D
39	0000128E	0E-19	1A	1B	1C	1D	1E	1F	20	21	21	22	22	23	23
40	000012A4	24-2F	30	31	32	33	34	35	36	37	37	38	38	39	39
41	000012BA	3A-45	46	47	48	49	4A	4B	4C	4D	4D	4E	4E	4F	4F
42	000012D0	50-5B	5C	5D	5E	5F	60	61	62	63	63	64	64	65	65
43	000012E6	66-71	72	73	74	75	76	77	78	79	79	7A	7A	7B	7B
44	000012FC	7C-87	88	89	8A	8B	8C	8D	8E	8F	8F	90	90	91	91
45	00001312	92-9D	9E	9F	A0	A1	A2	A3	A4	A5	A5	A6	A6	A7	A7
46	00001328	A8-B3	B4	B5	B6	B7	B8	B9	BA	BB	BB	BC	BC	BD	BD
47	0000133E	BE-C9	CA	CB	CC	CD	CE	CF	D0	D1	D1	D2	D2	D3	D3
48	00001354	D4-DF	E0	E1	E2	E3	E4	E5	E6	E7	E7	E8	E8	E9	E9
49	0000136A	EA-F5	F6	F7	F8	F9	FA	FB	FC	FD	FD	FE	FE	FF	FF
50	00001380	Sec 0035 00-0B	0C	0D	0E	0F	10	11	12	13	13	14	14	15	15
51	00001396	16-21	22	23	24	25	26	27	28	29	29	2A	2A	2B	2B
52	000013AC	2C-37	38	39	3A	3B	3C	3D	3E	3F	3F	40	40	41	41
53	000013C2	42-4D	4E	4F	50	51	52	53	54	55	55	56	56	57	57
54	000013D8	58-63	64	65	66	67	68	69	6A	6B	6B	6C	6C	6D	6D
55	000013EE	6E-79	7A	7B	7C	7D	7E	7F	80	81	81	82	82	83	83
56	00001404	84-8F	90	91	92	93	94	95	96	97	97	98	98	99	99

XI. 56 Reasons to Avoid Dungeons

		Level	DV	AV	Dam	SA%	SA#	SA Dm	Graph	Snd	M / N	M. Res	Spd	?	Mob	Neg	
#	Name	00	00	00	00	00	00	00	0	0	0	0	0	0	%	%	Notes
1	Goblin	01	01	02	02	00	00	00	F	E	A	0	2	2			
2	Kobold	01	01	01	02	00	00	00	F	0	E	0	2	2			
3	Rat	01	00	01	02	00	00	00	E	6	2	0	2	0			
4	Ooze	01	00	01	02	0A	09	01	0	7	1	1	2	0			
5	Evil Mane	01	00	02	02	0A	FF	01	F	0	A	1	2	0			
6	Giant Rat	02	01	01	03	00	00	00	E	6	2	0	2	0			
7	Lizard	01	01	01	03	00	00	00	2	A	7	0	2	0			
8	Imp	01	02	02	02	0A	0C	0A	9	F	E	2	2	2			
9	Orc	02	02	02	02	00	00	00	F	0	E	0	2	2			
10	Skeleton	01	02	02	03	00	00	00	6	1	2	1	2	0			
11	Wild Dog	02	01	01	04	00	00	00	2	d	3	0	2	0			
12	Spider	01	02	03	02	00	00	00	4	5	3	0	2	0			
13	Wolf	02	03	01	04	0A	FC	01	2	D	3	0	2	0			
14	Zombie	03	03	02	03	00	00	00	6	1	5	2	2	1			
15	Dark Slime	03	00	02	04	0F	05	01	0	7	1	1	2	0			
16	Spider	01	02	03	02	14	07	04	4	5	3	0	3	0			
17	Lemure	03	02	03	04	00	00	00	6	E	9	2	2	1			
18	Wight	02	03	03	04	00	00	00	1	C	A	2	2	0			
19	Dust Devil	02	02	FD	02	28	06	01	0	8	3	1	3	0			
20	Gremlin	01	04	FC	03	14	02	0A	F	F	A	2	2	2			
21	Cursed One	04	02	03	06	00	00	00	6	C	9	2	2	0			
22	Metalloid	03	04	03	04	14	09	02	F	A	2	0	2	0			
23	Snake	03	03	04	03	14	07	08	B	2	A	0	2	0			
24	Vampire Bat	02	04	03	03	0F	0A	01	A	0	3	1	3	0			
25	Smogg	03	06	03	03	32	0B	06	0	8	A	1	2	0			
26	Pixie	04	02	FC	04	05	FD	01	9	F	B	3	2	2			
27	Hobgoblin	04	05	02	06	05	FC	01	7	E	E	0	2	3			
28	Ghost	03	06	FB	04	14	0D	01	1	B	3	3	2	0			
29	Metazoid	04	04	04	04	1E	0F	04	8	A	E	1	2	0			
30	Land Crab	02	07	03	08	00	00	00	C	C	2	0	2	0			
31	Whiplash	03	04	04	04	28	ED	04	E	6	3	2	3	0			
32	Gnoll	04	05	05	06	00	00	00	F	B	A	0	3	4			
33	Troll	04	04	06	04	1E	0F	04	7	E	E	1	2	5			
34	Shambler	0A	01	04	02	0A	05	01	0	7	1	1	2	0			
35	Serpent	05	05	06	06	14	14	08	B	2	2	0	3	0			
36	Minotaur	06	06	05	06	00	00	00	8	D	A	1	2	4			
37	Ghoul	05	04	FA	04	05	FD	01	9	C	2	3	3	0			
38	Giant Wasp	04	05	06	06	14	08	0C	D	4	3	0	3	0			
39	Ogre	05	06	04	08	05	FC	01	7	1	A	0	2	5			
40	Devourer	06	06	07	04	1E	09	03	5	A	2	0	2	0			
41	Vampire	07	06	06	06	14	0A	04	1	3	A	4	2	5			
42	Scorpion	05	08	07	08	1E	08	0C	C	C	2	0	2	0			
43	Buzz Bomb	02	07	06	04	00	00	00	D	4	3	0	4	0			
44	Hill Giant	09	05	08	08	00	00	00	7	1	F	0	2	6			
45	Frost Giant	0A	06	07	0A	00	00	99	7	1	F	0	2	8			
46	Demon	07	07	F9	06	14	FE	0F	8	3	B	4	3	6			
47	Tarantula	09	07	09	06	1E	08	0C	4	5	3	0	2	0			
48	Wyvern	08	07	08	06	32	F2	08	3	9	7	2	2	4			
49	Demon King	0A	09	F8	08	1E	F3	03	8	3	B	6	2	8			
50	Dragon	0A	08	09	08	1E	F2	0C	3	9	F	3	2	6			
51	Elemental	05	0B	F9	08	1E	F0	04	9	8	7	5	3	0			
52	Pit Fiend	0A	09	F9	0A	14	FF	0C	1	3	B	6	2	A			
53	Arch-Devil	0C	0A	F8	0C	0A	EE	04	1	3	B	7	2	A			
54	Land Shark	0C	0C	09	0C	28	11	14	5	A	3	0	2	3			
55	Dragon Lord	0C	09	0A	0A	32	F2	0C	3	9	F	4	3	8			
56	Will O'Wisp	05	0C	04	06	0A	FD	01	9	B	3	5	5	0			

XI. 56 Reasons to Avoid Dungeons

Monster Combat Queue – This Queue is updated each time Monsters are encountered and potential action is imminent.

Game Byte #	Sector 002F Bytes	Monster Information
00000DA6	26-31	Monster Name
00000DB2	32	Level
00000DB3	33	DV
00000DB4	34	AV
00000DB5	35	Damage (Max Damage)
00000DB6	36	Special Attack Chance %
00000DB7	37	Special Attack Type (from list)
00000DB8	38	Special Attack Damage
00000DB9	39	Monster Sound Table Pointer
00000DBA	3A	Monster Graphic Pointer
00000DBB	3B	Negotiation (0 - 3)
00000DBC	3C	Mobility (1 - 4)
00000DBD	3D	Monster Magical Resistance
00000DBE	3E	Monster Luck? (Last hex # of Monster Attributes is placed here???)
00000DBF	3F	Monster Speed
00000DC0	40	Hit Points Monster #1
00000DC1	41	Hit Points Monster #2
00000DC2	42	Hit Points Monster #3
00000DC3	43	Hit Points Monster #4
00000DC4	44	Hit Points Monster #5
00000DC5	45	Hit Points Monster #6
00000DC6	46	Hit Points Monster #7
00000DC7	47	
00000DC8	48-49	Monster #1 Location
00000DCA	4A-4B	Monster #2 Location
00000DCC	4C-4D	Monster #3 Location
00000DCE	4E-4F	Monster #4 Location
00000DD0	50-51	Monster #5 Location
00000DD2	52-53	Monster #6 Location
00000DD4	54-55	Monster #7 Location
00000DD6	56-57	

Only Spells that Increase or Decrease a Monster's HP affect an individual Monster, which is recorded here. All others affect the group.

Other Game Settings

00000DD8	58 (05)	
00000DD9	59 (0A)	
00000DDA	5A (0F)	
00000DDB	5B (03)	
00000DDC	5C (0E)	Lowering # decreases Party Combat Probability [1]
00000DDD	5D (0A)	Lowering # increases Party Combat Probability [2]
00000DDE	5E (00)	
00000DDF	5F (00)	
00000DE0	60 (1E)	% Probability of Hearing Monster when pressing "L" [3]
00000DE1	61 (28)	% Prob. of not getting wounds at Vaults [4]
00000DE2	62-71	Word "Combination" - 16 bytes

Note: Monster's sound and graphic pointers are in reversed order in Queue (compared to list) presumably because you can hear them before, when pressing 'L', or as you enter the room, before you see them.

1. All else being equal, possibly baseline Party Luck
2. All else being equal, possibly baseline Monster Luck
3. When pressing L for Listening at door
4. i.e. when guessing wrong at Vault combination

XII. A 'Party' Should be Fun – Creating Characters

Whether in a book, movie or computer adventure, character development is crucial to the storyline. If the characters appear humdrum then the entire story will likely be as well. A 10-level TOD adventure is lengthy for a TI99/4a, so every opportunity to grasp and maintain player interest must be employed. Designing unique, interesting characters is critical in meeting this goal. In Tunnels of Doom it is possible to create and populate your Party with up to four unique Classes of Characters.

The ***Tunnels of Doom*** manual aptly defines Class as: "Defines a Player's skills and the limitations of his trade." To date, a better definition of a TOD Class has not been forthcoming. Thus the task now before us will be to define our four Classes by; making them unique enough, useful enough and useless enough (at the same time!) In short, (in general) all "Classes" should have their drawbacks as well as strengths to encourage a Party or group effort play.

Quest achieved this goal by providing two pathways of play. For a Party of characters, three unique Classes were devised:

- Fighter – Can use all weapons, shields and armor. Can use all Magical Items except for Scrolls.
- Wizard – Limited in use of weapons and armor. Can use all Magical Items including Scrolls.
- Rogue – Has limited armor use, but can use all weapons. Can use all Magical Items except Scrolls. In addition, the Rogue has unique "trap avoidance" abilities.

A player can select up to four characters from these three Classes to form his Party. Additionally, a fourth type of Character is selectable for a one Party (Player) game:

- Hero – Can use all weapons, armor and shields. Can use all Magical Items including Scrolls and has the Rogue's trap avoidance abilities.

For the newbie, a single "Hero" type character can enhance survival and achievement of Quest objectives. Let's look at how a TOD Class, "Defines a Player's skills and the limitations of his trade." Amazingly, for each Character Class, definition is achieved through the modification of a single byte of memory!

With each of the four possible Classes the first hex digit of the Class definition byte defines the "Player's skills", while the second hex digit "the limitations of his trade. These are summarized as follows:

1st hex digit defines Class Skills:

HexHex

0 = no special skills
1 = can use all Magical Categories
2 = avoids traps (chests & vaults)
3 = uses all Magical Categories
and avoid traps
(**4 - F** repeat this sequence)

A= All. **L**= Limited

The underscored Hex values may be used without any special consideration in your game database.

See Limitations Key below

2nd hex digit Class limitations, (in weapon/ armor use):

	Hand	**Ranged**	**Armor**	**Shield**
0 =	**L**	**L**	**L**	**L**
1 =	**L**	**L**	**L** [1]	**L** (repeats 0)
2 =	**L**	**L**	**L**[2]	**A**
3 =	**L**	**L**	**A**	**A**
4 =	**A**	**L**	**L**	**L**
5 =	**A**	**L**	**L**[1]	**L** (repeats 4)
6 =	**A**	**L**	**L**[2]	**A**
7 =	**A**	**L**	**A**	**A**
8 =	**L**	**A**	**L**	**L**
9 =	**L**	**A**	**L**[1]	**L** (repeats 8)
A =	**L**	**A**	**L**[2]	**A**
B =	**L**	**A**	**A**	**A**
C =	**A**	**A**	**L**	**L**
D =	**A**	**A**	**L**[1]	**L** (repeats C)
E =	**A**	**A**	**L**[2]	**A**
F =	**A**	**A**	**A**	**A**

Key: A= All. **L**= Limited, i.e. items are not generally available, but will be if entered in reverse notation. **L¹** = Limited, except for the <u>first</u> Armor type, which is always available. **L²** = Limited, but the <u>Store</u> will let you purchase items 2-8, take your money, and then say "You can't use those" – so you end up with nothing. **L¹** & **L²** are apparent programming glitches. **L²** items are only a concern when Stores are present and the item is for sale. If these items are marked as "only found in Dungeon", then there is no concern with use of this Hex digit.

It is likely that the original intent was to use the 2nd Hex Byte to define every combination of All & Limited use possible for each category (using 16 hex digits, 0 - F), but matters did not work out that way. On the preceding table underlined second Hex digits can <u>always</u> be used without concerns in game play. For the rest conditions apply, as noted above and the game creator must anticipate these if they are used in defining a Class. It is apparent that many Classes/ types of Characters are possible using just one Byte!

Quest for the King uses: 0F = Fighter (no special skills, all weapons), 10 = Wizard (use scrolls, limited weapons/ armor), 2C = Rogue (avoids traps, all weapons/ limited armor), 3F = Hero (uses scrolls, avoids traps, all weapons/ armor). Remember, <u>any</u> of the 8 Magical Categories may be restricted in use to only Classes with hex 1 or 3 as their first hex attribute digit,

<u>Other examples</u>:

Thief	= <u>24</u> = Avoids Traps, uses all Hand Weapons; limited Ranged, Armor, Shields. (Rogue, Thief, Scoundrel,)
Archer	= <u>08</u> = no special Ability, uses all Ranged; limited Hand, Armor, Shields
Archer	= <u>0A</u> = no special Ability, all Ranged & Helmets (replaces Shields); Limited Hand/ Armor
Scientist	= <u>30</u> = uses all Magical Categories & Avoids Traps; limited in all Weapons/Armor. (Tech, Alchemist, Wizard)
Dwarf	= <u>27</u> = Avoids traps, all Hand, Armor, Shields; Limited Ranged.
Elf	= <u>18</u> = uses Scrolls, all Ranged; limited Hand, Armor & Shields. (Useful in 2nd row behind fighters)
Healer	= <u>13</u> = Scrolls, Armor, Shields, Limited Weapons. (Healer, Medic, Elf, Shaman, etc.)
Hobbit	= <u>28</u> = Avoids Traps, all Ranged Weapons (bow) Limited Hand Weapons, Armor, Shields.
Robot	= ??? be creative!

In addition to the two hex digits that define Skills and Limitations, development of a unique Class is enhanced by assigning other specific elements to the Class, e.g.: <u>Gold</u> amount at game start, <u>Hit Points</u>, special & unique <u>Weapons</u> or <u>Armor</u> (e.g. Mithril Mail, Staff of Power, RPG) and useful <u>Magical Items</u>. Except for Class Hit Points, these elements are defined in Sector 002F, which is used as the initial starting point for each Player Class in every "New Game":

Initial Values for Player Classes (Status) Defined (Class Stats with: "New Game" Option)

Sector	1ST CLASS		2ND CLASS	3RD CLASS		4TH CLASS		
002F	Game Byte #	(002F) Bytes	Game Byte #	(002F) Bytes	Game Byte #	(002F) Bytes	Game Byte #	(002F) Bytes
Class Name	00000DF2	72-7B	00000E08	88-91	00000E1E	9E-A7	00000E34	B4-BD
Gold to Start	00000DFC	7C	00000E12	92	00000E28	A8	00000E3E	BE
Class Abilities	00000DFD	7D	00000E13	93	00000E29	A9	00000E3F	BF
Armor (list)	00000DFE	7E	00000E14	94	00000E2A	AA	00000E40	C0
(00)	00000DFF	7F	00000E15	95	00000E2B	AB	00000E41	C1
Weapon #1 (list)	00000E00	80	00000E16	96	00000E2C	AC	00000E42	C2
(00)	00000E01	81	00000E17	97	00000E2D	AD	00000E43	C3
Weapon #2 (list)	00000E02	82	00000E18	98	00000E2E	AE	00000E44	C4
(00)	00000E03	83	00000E19	99	00000E2F	AF	00000E45	C5
Magical Item #1	00000E04	84	00000E1A	9A	00000E30	B0	00000E46	C6
(00)	00000E05	85	00000E1B	9B	00000E31	B1	00000E47	C7
Magical Item #2	00000E06	86	00000E1C	9C	00000E32	B2	00000E48	C8
(00)	00000E07	87	00000E1D	9D	00000E33	B3	00000E49	C9

Note: It would seem logical that 7F=Shield Type, 81 & 83=Ammo amount if a ranged weapon in 80 & 82, but values entered here do not appear to do anything.

The assigned values for this table in ***Quest*** are as follows:

`Sector 002F` ***Initial Attributes by Player Class (Status) Defined*** (Class Stats with: "New Game" Option)

Game Byte #	Initial Attributes	Sector Byte	1st Class Name	Sector Byte	2nd Class Name	Sector Byte	3rd Class Name	Sector Byte	4th Class Name
00000DF2	Class Name (Status)	72-7B	Fighter	88-91	Wizard	9E-A7	Rogue	B4-BD	Hero
00000DFC	Gold to Start	7C	0A	92	05	A8	08	BE	0C
00000DFD	Class Abilities	7D	0F	93	10	A9	2C	BF	3F
00000DFE	Armor (from list)	7E		94		AA		C0	
00000DFF	(00)	7F		95		AB		C1	
00000E00	Weapon #1 (list)	80		96		AC		C2	
00000E01	(00)	81		97		AD		C3	
00000E02	Weapon #2 (list)	82		98		AE		C4	
00000E03	(00)	83		99		AF		C5	
00000E04	Magical Item #1	84		9A	11	B0		C6	1A
00000E05	(00)	85		9B	04	B1		C7	06
00000E06	Magical Item #2	86		9C		B2		C8	
00000E07	(00)	87		9D		B3		C9	

The creator of the ***Quest*** database expected bytes 85, 9B, B1, C7, etc. to express the maximum number of uses for the Spell in question. However, the number of uses actually defaults to the hex value provided in the List of 40 Magical Items. In the same spirit, it would seem that for Class #1, bytes 7F=Shield Type, 81 & 83=Ammo amount if a ranged weapon in 80 & 82, but any values entered here are also ignored. It would appear that here once again, memory limitations of the game program proper left this game database feature unimplemented.

The amount of Gold to Start is valid for the Easiest game option. Selecting Medium will reduce the Gold amount to 3/4 (rounded upwards) and the Hardest option reduces the amount by ½. The hex values for Armor, Weapons and Magical Items represent their ordinal placement from their respective Lists including any blank fields. Thus, any hex value >08 will be a Ranged Weapon (from the Weapons List) and any >08 will be a Shield (from the list of Armor types). A **Class** can be equipped with both a Hand and Ranged Weapon at the start of a new game, but only Armor ***or*** a Shield.

`Sector 002F` ***Initial Attributes by Player Class*** `from SUB-Level 12`

Game Byte #	Initial Attributes	Sector Byte	1st Class Name	Sector Byte	2nd Class Name	Sector Byte	3rd Class Name	Sector Byte	4th Class Name
00000DF2	Class Name (Status)	72-7B	Weapon-SPC	88-91	Researcher	9E-A7	Security	B4-BD	
00000DFC	Gold to Start	7C	00	92	00	A8	00	BE	
00000DFD	Class Abilities	7D	1F	93	28	A9	0F	BF	
00000DFE	Armor (from list)	7E	03 Combat Vest	94	00	AA	00	C0	
00000DFF	(00)	7F	00	95	00	AB	00	C1	
00000E00	Weapon #1 (list)	80	0F FN P90	96	0D Taser	AC	0C Mossberg	C2	
00000E01	(00)	81	00	97	00	AD	00	C3	
00000E02	Weapon #2 (list)	82	04 Ka Bar	98	00	AE	09 Glock 17	C4	
00000E03	(00)	83	00	99	00	AF	00	C5	
00000E04	Magical Item #1	84	15 IFAK	9A	16 Suture Kit	B0	15 IFAK	C6	
00000E05	(00)	85	00	9B	00	B1	00	C7	
00000E06	Magical Item #2	86	1E SMAW II	9C	18 Medic Bag	B2	04 Wafer	C8	
00000E07	(00)	87	00	9D	00	B3	00	C9	

In the preceding table:

- The Weapon-SPC has 1F for Class Abilities. 1=can use all Categories of Magical (Tech) Items including those restricted for use (*SMAW* and *L-II Weapon*). F=no restrictions in weapons or armor usage.
- Researcher has 28 for Class Abilities. 2=trap avoidance skills. 8=limited in use of hand weapons, armor and shields (Helmets in this scenario).
- Security has 0F for Class Abilities. 0=cannot use the SMAW or L-II Weapon Magical Categories (here, Tech Categories). F=no restrictions in weapons or armor usage.

The table of Initial Attributes provides an opportunity to provision a specific Class (or Classes) with a *unique* Weapon, Armor or Shield, which otherwise could neither be purchased nor found within the Dungeon. ***Halls*** employed this feature to equip the Wizard with a *Wizard Staff* and the Hobbit with *Mithril Mail* Armor. In a similar fashion ***Sub12*** equipped the Researcher with a *Mark-30 Taser*. As both the *Wizard Staff* and *Taser* were formidable, their use was restricted to “Once/ Battle”. ***Sub12*** also assigns the Weapon-SPC with an *FN P-90*, which otherwise could only be acquired (found) in Bunker Levels 8 through 12. See previous table for details.

You may also supply up to two Magical Items, including those that would typically only be found deep within your dungeon. Again, using ***Sub12*** as an example, the Weapon-SPC is equipped with an *SMAW II Serpent*. See previous table for details.

Two other options for use in defining unique Classes are currency and Hit Points. Gold or any designation you may choose to use for currency in your game is extremely helpful in equipping your Party. In ***Halls*** our Dwarf was provided with 400 Gold coins, which he shared with the Party – of course! The initial values for Class Hit Points are located in **Sector 0044**:

```
Game        Sector 0044
Byte #      Bytes
00002310    90-93  Initial HP for the 4 Classes. Class #1 in byte 90, #2 in byte 91, #3 in byte 92 and #4 in byte 93.
```

Permitting the option for a Hero Class of character, who is only available as a game choice in a one Party game, is accomplished by a single byte in Sector 003A:

```
Game        Sector 003A
Byte #      Bytes
00001961    E1     (FC) Sets the number of available Party Classes.
                   Accepted vales: (FF - FC or 1 - 4) Use of Reverse Notation causes the last Class to be available
                   only in a 1 Player Game (as in Quest).
```

Note – This will only set the option for a Hero type Player. It does not assign any Class skills or limitations, which must be assigned in the Class attribute byte as for any other Class. ***Quest*** assigns `hex 3F` for this Class, but this can be modified as desired.

Class (Character) Graphic representations are defined in **Sector 002F**. These *pattern-identifier* hex strings are available anytime the game player selects a different number of Party members than the last saved game, or, opts to Redo the Party Characters and their Names when this option is made available in the TOD start-up Screens.

To insure that Class/ Character game graphics are as you intended you must also redefine the *pattern-identifier* hex strings in **Sector 0026**, which is updated each time a game in progress is saved. These strings should be redefined or cleared to prevent potential conflicts.

Graphics Associated with the Classes or Character Types

Game	Sector 002F	
Byte #	Bytes	
00000E4A	CA-E9	1st Class - Defensive Graphic
00000E6A	EA-09	1st Class - Attack Graphic (to Sector 0030)2FSector 0030
00000E8A	0A-29	2nd Class - Defensive Graphic
00000EAA	2A-49	2nd Class - Attack Graphic
00000ECA	4A-69	3rd Class - Defensive Graphic
00000EEA	6A-89	3rd Class - Attack Graphic
00000F0A	8A-A9	4th Class - Defensive Graphic
00000F2A	AA-C9	4th Class - Attack Graphic

The *character-codes* for these Class graphics are assigned when/ if this Class is selected for play in a game. They are then relocated in VDP memory to its disk counterpart in Sector 0026 when saved to disk with each saved game. Their order here is determined by their selection in the game setup, or, that defined by the last **O**rder change during game play.

Saved Character Graphics – Common Graphics Bank (common to all three >7F Banks)

Game	Sector 0026 (Defines up to 4 Players in Current or last Saved Game)		
Byte #	Bytes		Char-Codes
00000480	00-1F	Character #1 Defense pose	00-03
000004A0	20-3F	Character #1 Attack pose	04-07
000004C0	40-5F	Character #2 Defense pose	08-0B
000004E0	60-7F	Character #2 Attack pose	0C-0F
00000500	80-9F	Character #3 Defense pose	10-13
00000520	A0-BF	Character #3 Attack pose	14-17
00000540	C0-DF	Character #4 Defense pose	18-1B
00000560	E0-FF	Character #4 Attack pose	1C-1F

Note: These Character graphics are used if you select, "Continue Current Game" or, they are offered for use if the number of Players selected is the same as those of the last saved game.

XIII. Descent into Dank, Dark, Dangerous, Depths – Designing Dungeons

Designing original ***Tunnels of Doom*** databases can be ***very*** complex indeed. In planning for this undertaking it is helpful to view each database as comprising three distinct steps or phases:

1. **Graphics -** Graphic design of your dungeon. This critical step provides a game with a unique look, feel and character. Unfortunately, the only two game databases produced by *Texas Instruments*, ***Quest for the King*** and ***Pennies***, used the same Dungeon design. Even thirty years later there are only a few other examples. In addition to creating unique Dungeon design graphics, there remains the entire gamut of characters, monsters, weapons etc. to be considered.
2. **Dungeon Layout -** Once the graphics have been planned, and are hopefully well into their development stage, there is the Dungeon layout to be considered. In this instance we had the good fortune that T.I. provided <u>two</u> unique examples of dungeon layouts. This provided confirmation that this was an area in which the game database developer could exercise some measure of creativity. Quest has up to 10 levels of play, 20 rooms per floor, 2 hallway fountains and a host of items. Pennies has a maximum of 4 floors, 12 rooms per floor, no hallway fountains (or monsters), etc. In short, the very nature of their differences provided insight into what could be manipulated within a game database.
3. **Global Settings -** The last major category of Dungeon design takes into account a wide array of global settings, which when manipulated produce additional game uniqueness.

Graphics

Hopefully you have already given some thought to your intended game graphics. This is typically the most difficult and labor intensive portion of any <u>new</u> dungeon. I would suggest that your game be themed and that all graphic generation; Players, Monsters, Items/ Weapons and the Dungeon itself, attempt to reflect your theme. There are numerous TOD game databases available, which you can consult for ideas.

For dungeon design proper there are a few examples. I have already mentioned ***Halls*** and ***Sub12***. A third, and very interesting one at that, is titled, ***The Forest of Ruin.*** This game base takes an entirely original and refreshing approach to "Dungeons", their "design" and setting, making it an excellent example of what can be done. Experimentation will be required to get desired results. However, a few general suggestions might prove useful.

- The enjoyable, 3-D perspective of ***Tunnel of Doom*** also makes it somewhat difficult to design around. To create this effect 8x8 pixel graphic blocks are removed from both hallway side as you look down the dungeon hallway. This gives the perspective of hallway depth. Similarly, when a door is viewed to your left or right side, there is one additional 8x8 pixel graphic block on the proximal portion of the door versus its most distal portion. This creates a bizarre effect when horizontal graphics are used and viewed from the side (directly ahead is generally fine).
- There are three basic techniques to get past this issue. The simplest, used by the original TOD game designer, is to leave the walls <u>blank</u> and rely on hallway coloration for a unique look. The second, also used in ***Quest*** for door design, is to primarily employ <u>vertical</u> designs. This helps to eliminate the skewed appearance when viewing hallway features to either side. This holds true for both hallways and doors. A third method is to use carefully planned 8x8 pixel block designs for doors and walls. ***Halls*** provides an example for this technique.
- Only after a decision has been made on general wall design can you proceed to the several necessary interfaces: wall/ ceiling, wall/ floor, door/floor, etc.
- Because the hallway coloring scheme often relies on the background color of one hallway set, e.g. walls, being the foreground coloring of another set, e.g. ceiling, it is often prudent to plan on the ceiling and floors, or the walls, on being "plain". It is challenging to have specific designs for the walls ***and*** ceiling and getting the colors to cooperate.
- Early on, you will also have to decide if you want to have (the equivalent of) hallway "Fountains" or "Statues" as their designs may be quite different and will require early planning.
- For all of this, practice will make perfect!

Dungeon Layout

The Tunnels of Doom program proper is rather forgiving in a number of respects. If a database designer's request is not possible, is too complicated, has it running its algorithms too many times, or for too long a time, it frequently just moves on. This is particularly true in matters of dungeon construction.

Quest – and any game that used it as its basis, was designed to accommodate 10 levels (floors) of play; with 20 rooms per floor, two sets of stairs up and down and two hallway fountains. There is also the requirement that the minimum number of floors is one. If a game designer is content to work within these parameters there is nothing additional to consider. If you desire to decrease any of these parameters for your specific game, then it is a matter of changing a few hex digits in **Sector 003A**:

Game	Sector 003A	
Byte #	Bytes	Default/ Controls
00001962	E2	(0A) Maximum # of Floors.
00001966	E6	(02) # of Fountains or Statues in Hallways. (See Byte E9 to set as Fountain or Statue)
00001967	E7	(02) # of Stairways down and up per Floor.
0000196C	EC	(14) # of Rooms/ Floor
0000196D	ED	(01) Minimum # of Floors in Game

The number of Floors, Fountains, Stairways and Rooms for your specific design can be reduced by decreasing the hex digit in the appropriate bytes. The minimum number of Floors (e.g. in ***Halls***) can be increased by increasing the hex in byte ED. However, considerably more is possible if you are willing to do a little math.

The database reserves up to 2,140 bytes of VDP memory for dungeon design. If your intended game will "fit" into that amount of memory – then it probably can be done! However, once again, some conditions apply and some simple guidelines will enable you to calculate dungeon memory usage.

- Dungeons have a **maximum of 2,140 Bytes** available for their design.
- Each room requires 10 bytes of memory
- Each Stairway Down requires 2 bytes of memory
- Each Stairway Up requires 2 bytes of memory
- Each Hallway Fountain requires 3 bytes of memory. However, note the following:
  - For an even number of Fountains per Floor:
    - Bytes = Fountains x 3
  - For an odd numbers of Fountains per Floor:
    - Bytes = (Fountains x 3) +1 so 3 Fountains will require 10 bytes of memory

The sum total of bytes (per floor) must be computed and then multiplied by the intended maximum number of floors for the game. This result should **total 2,140 Bytes** or less. Using ***Quest*** as our example we calculate the following:

- 20 Rooms per Floor = 200 bytes of memory
- 2 Stairways Up and Down (the numbers must coincide) = 8 bytes of memory
- 2 Hall Fountains = 6 bytes of memory

Our total is: 214 bytes per Floor x 10 (possible) Floors = **2,140 bytes** and all is well! Our next step is to let the Tunnels of Doom program know what we expect from it. These are set in **Sector 003B**:

Game	Sector 003B		
Byte #	Bytes	Default/ Controls	
00001984	04	(00)	
00001985	05	(D6) Total # of Bytes reserved for Rooms, Stairs & Hall Fountains	(hex D6=16*13+6=214)
00001986	06	(00)	
00001987	07	(C8) Total # Bytes reserved for Rooms (10 Bytes/ Room reserved)	(hex C8=16*12+8=200)
00001988	08	(00)	
00001989	09	(CC) Total # of Bytes reserved for Rooms & Stairs down	(hex CC=16*12+12=204)
0000198A	0A	(00)	
0000198B	0B	(D0) Total # of Bytes reserved for Rooms, Stairs down & Stairs up	(hex D0=16*13=208)

The difference between the first calculation, 214 and the last, 208 is the number of bytes reserved for Hall Fountains = 6 bytes. What appears to happen in real time is something like this:

1. 214 bytes are set aside in VDP memory for each Floor. Floor number is selected at setup.
2. The first 200 bytes are allotted for Rooms.
3. Four more bytes to stairs down.
4. Another 4 bytes for stairs up
5. The remainder (6 bytes) are set aside as Fountain bytes.

If only understanding memory use and management on the SAMS card was this easy! When a game developer selects fewer Rooms, etc. per Floor than the default values for Quest, the extra bytes are left "00". The TOD program is very good at generating "something" during dungeon creation, even when you input the hex numbers incorrectly. Even under the best of circumstances, dungeon generation will, now and then, produce an off result. This will typically show up as a missing room. Failure to keep a dungeon to 2,140 bytes will result in overwriting of other, necessary, portions of the database.

In addition to these "memory management" settings, the previously referenced values in **Sector 003A** must be set to agree with your management scheme. What about a 12 Level dungeon? ***Sub Level-12*** uses the following settings: 12 Floors, 16 Rooms/ Floor, 2 Stairs up & down, 1 Hall "Statue" which add to:

16*10+4+4+4=172 bytes per Floor, 12*172=**2,064** total dungeon bytes.

Global Settings

Now that the dungeon graphics are complete and your floor plan is in place, it is time to populate the dungeon with valuables and other item of interest. We have seen how weapons, armor, magical items and the like were assigned floor distribution by defining a starting floor in their respective Lists. From there, items are randomly assigned by the program. Quest Items are the exception and these can be assigned to a specific floor or a range of floors. In addition to providing the program with guidance on floor distribution the database can also provide input in assigning quantity levels. This is also done on a floor by floor basis.

Floor Information* - *10 Bytes of data/ Floor to set limits on the Quantities of various Room Items

Game	Sector 0044		Game	Sector 0044	
Byte #	Bytes	Quantity Information	Byte #	Bytes	Quantity Information
00002314	94-9D	Floor 1 Information	00002346	C6-CF	Floor 6 Information
0000231E	9E-A7	Floor 2 Information	00002350	D0-D9	Floor 7 Information
00002328	A8-B1	Floor 3 Information	0000235A	DA-E3	Floor 8 Information
00002332	B2-BB	Floor 4 Information	00002364	E4-ED	Floor 9 Information
0000233C	BC-C5	Floor 5 Information	0000236E	EE-F7	Floor 10 Information

Should more than 10 Floors be desired, this Table of Floor Information Bytes will need to be manually extended in a game database. See ***Sub12*** for details. 10 bytes of room quantity information are assigned for each floor. Their significance is as follows:

Floor Information - ***10 Bytes of data/ Floor to set limits on the Quantities of various Room Items***
Game **Sector 0044**
Byte # **Bytes** for Floor #1

00002314	94	95	96	97	98	99	9A	9B	9C	9D
Quantity	**Gold**	**MI**	**Weapons**	**Armor**	**Maps**	**?**	**Statues**	**Fountains**	**Chests**	**%**
Quest Value	28	FB	FF	FE	FE	00	FF	FF	02	0A

Gold – probability that a room will contain gold.	Hex 28=40%
Magical Items – number of Magical items.	Hex FB=5
Weapons – number of Weapons on floor.	Hex FF=1
Armor – number of Armor.	Hex FE=2
Maps – number of Maps.	Hex FE=2
? – yet another unknown.	Hex 00
Statues – number of Room Living Statues.	Hex FF=1
Fountains – number of Room Fountains.	Hex FF=1
Chests – Chest quantities. (see below)	Hex 02=a random distribution
Chests % – probability that a Chest will have a trap.	Hex 0A=10%

The Floor Quantity Information table uses Reverse Notation for amounts, to exact specific quantities. Regular hex numbers such as "01" would only set a maximum amount. It would be unfortunate if "02" as a maximum amount did not provide the required Map necessary to proceed to the next Floor!

There are practical limits to room quantities as three items (total) from Magical Items, Weapons and Armor appears to be the limit for a room. I have reliably been able to populate up to 20 (EC) Magical Items and 10 (F6) Weapons and Armor per floor (***Sub12***), which may be close to the limit for a 16 room dungeon. The number of rooms per floor is a factor, as are rooms reserved for Stores and Vaults. Do not include rooms reserved for Stores, or empty Vault rooms seeded with items in the quantity distribution. A reasonable rule of thumb per floor is (rooms * 3 - (Maps + Quest Items + Stores + 6)) for the total limit, with one Armor and Weapon maximum per room. As you near quota limits for all 3 categories you will observe greater success with lower prefixed items - see Chapter XV for details.

The 6th Floor Quantity byte is set to "00" on all floors in Quest. Placing a value here, e.g. FF=1 places hex "04" where a room typically stores Chest information, but it does not appear to be otherwise functional. I unfortunately have not had the opportunity to explore this further.

Values for Chests are as follows: 00 = every room has a Chest.
01= every room that has an item will have a Chest.
02 = Chests are randomly distributed in rooms.
03 = no Chests (hex 03 and >)

Using a hex 00 value for Chests will force the appearance of a Chest in a room that would otherwise be reserved only for Stores and Vaults. This will, of course, only work if neither is actually present.

Some items cannot coexist with others. Chests, Statues and Fountains will not be found in rooms with Vaults. Statues and Fountains will not be in the same room. If there is not enough "room" in your rooms for your quantity designations, then a left to right precedence generally takes place. If you have 5 Rooms reserved for Stores and Vaults, set a value for 15 Statues, there will be no room left for Fountains on a 20 room floor. Quest items assigned to a specific floor will take precedence over all other items during distribution. The distribution of all items, monsters etc. to their respective rooms, takes place during the creation of a new game – while we are enjoying the ***Tunnels of Doom*** theme song.

This chapter will discuss a potpourri of game options that will affect your game database on a global level. Hence, common sense and some appreciation of the big picture is necessary. A number of these options have already been considered in previous sections, but will again be listed here for reference. `Sector 003A`, which is primarily used to assign global options, is a good place to start.

Global Game Settings for New/ Current Game

Game	Sector 003A	
Byte #	Bytes	Default/ Controls
0000195A	DA	(04) Max # of Players (1–4)
0000195B	DB	(0A) # of Rations/ unit purchased
0000195C	DC	(00) (When a # is placed here, # of Rations never changes.) Writes to Sector 002F, Byte 16.
0000195D	DD	(02) Baseline Ration Consumption Interval. 02 = 1 Ration consumed every 20 paces. Writes to Sector 2F, Byte 17.
0000195E	DE	(02) Cost of Rations per Purchase. (x Factor)
0000195F	DF	(03) Paces for Wandering Monster Probability check. Writes to Sector 002F, Byte 12.
00001960	E0	(02) Baseline Party Healing Interval. Writes to Sector 002fF, Byte 19.
00001961	E1	(FC) Number of Party Classes. (FF-FC or 1-4) Reverse notation = last Class only in 1 Player Game.
00001962	E2	(0A) Maximum # of Floors.
00001963	E3	(28) \ E3 x E4 = # of Experience Points required to advance a Level. (E3 x 10 = for 1st Level)
00001964	E4	(0A) / (E4 = Increment Factor to advance to next Level)
00001965	E5	(14) # of Ammo bought per purchase. (This entry will override any ammo quantity in the Ranged Weapon list if <E5.)
00001966	E6	(02) # of Fountains or Statues in Hallways. (See Byte E9 to set as Fountain or Statue)
00001967	E7	(02) # of Stairways down and up per Floor.
00001968	E8	(01) Map Status: (see note)
00001969	E9	(07) 07="Who will use the Fountain?" 08="Who will use the Statue?" (Enables Hallway Statues.)
0000196A	EA	(03) # of Players in Current or/Saved Game (used in restarting game/ restocking dungeon, # of Players in New Game is same)
0000196B	EB	(02) # of Floors Selected (also, default value if Restocking Dungeon or Continue a Current Game)
0000196C	EC	(14) # of Rooms/ Floor
0000196D	ED	(01) Minimum # of Floors in Game
0000196E	EE	(03) Max # of Vaults & Stores per Floor (total of both - max value = 05)
0000196F	EF	(0A) Gold for Statues (x factor) (1st Floor)
00001970	F0	(00)
00001971	F1	(00)
00001972	F2	(00)
00001973	F3	(3C) (60%) Probability of Hallway Monster attack after each Pace counter countdown is complete. (see Byte DF)
00001974	F4	(02) # of Floors Selected in Game
00001975	F5	(02) # of Floors Selected in Game
00001976	F6	(03) # of Players in Current Game (probably for gold to start with calculation)
00001977	F7	(00) Game Difficulty Selected: 00=Easiest, 01=Medium, 02=Hardest
00001978	F8	(00) Current Floor
00001979	F9	(00)
0000197A	FA	(00) Current Player in Action Queue (01-04), 05 = end of all individual player turns, 00=moving as a Party
0000197B	FB	(01)
0000197C	FC	(06) Max amount of Healing that occurs when Rations are Consumed
0000197D	FD	(0A) Cost of Healing x factor (Purchased at Stores)
0000197E	FE	(1E)
0000197F	FF	(06) Max amount Player's HP can increase with new Level.

While many of these options reflect a player's game selections at startup or a game's ongoing status, a number of them can be modified to create custom game scenarios. Some of these values are immediately written to `Sector 002F`, which primarily reflects the moment by moment game status (and hence a saved game). There are also a few useful default settings, reviewed in the last section on Game Scenarios.

Notes: The 'Factor' is usually the Current Floor level (in hex) value, or that set in Sector 0050. See this Chapter for details.

`Byte DA` - can be a number from 1-4 to indicate the maximum number in the Party.

`Bytes DC, DD, DF & E0` - are game defaults written to `Sector 002F`. The values in `Sector 002F` may be temporarily altered by Spells, these values reset the baseline once the Spell expires, or with each new game. These values represent paces or key presses (x10 or another factor)

`Byte E1` - can be a number from 1-4, or FF-FC, to indicate the maximum number of Player Classes available in a game. If reverse notation is used, the last Player Class, e.g. 'Hero', will only be available in a one Player game. See Chapter XIII.

`Byte E5` - minimum ammo quantity, a value entered here will overwrite any ammo amount set in the Ranged Weapon List, if the latter is less than `E5`. Plan accordingly! (Appears to be a glitch.)

`Byte E6` - this will set the number of Hallway Fountains, or Statues. `Byte E9` must be set to determine which will be used. The correct number of Bytes must also be set in `Sector 003B`.

`Byte E7` - this will set the number of Stairs down & up per floor. The correct number of Bytes must also be set in `Sector 003B`.

`Byte E8` Map status: Hex 00 = map unnecessary to descend, floor is always visible.
Hex 01 = map necessary to descend, only explored sections visible without map.
Hex >01 = map unnecessary to descend, explored rooms are visible, but not hallways without map.

Note: Changing the Map color bytes can make Unexplored Map sections visible as well.

`Byte E9` 07="Who will use the Fountain?" 08="Who will use the Statue?" Enables Hallway Statues. Other value do not appear to have an effect.

`Byte EC` The correct number of Bytes must also be set in `Sector 003B`.

`Byte EF` Living Statues require the Party to have some Gold before they will function. If this is set to 00, then no gold will be taken to use statues. However, Gold is necessary to initiate the dialogue.

Global Game Settings of New/ Current Game (concluded)

Game	Sector 003B	
Byte #	Bytes	Default/ Controls
00001980	**00**	(06) Current Party Setting: 01=Hallway, 02=Rooms with possible contents, 03=Hallway Fountain, 04=Room with Stairs Down, 05=Room with stairs Up, 06=Store (Apparently for graphic cues)
00001981	**01**	(02) If changed, resets back to'02'.
00001982	**02**	(00) Direction the Party is facing: 00=North, 01=East, 02=South, 03=West
00001983	**03**	(00) (60=monsters in room – uncertain of other meanings - dynamic value)
00001984	**04**	(00)
00001985	**05**	(D6) Total # of Bytes reserved for Rooms, Stairs & Hall Fountains
00001986	**06**	(00)
00001987	**07**	(C8) Total # Bytes reserved for Rooms (10 Bytes/ Room reserved)
00001988	**08**	(00)
00001989	**09**	(CC) Total # of Bytes reserved for Rooms & Stairs down
0000198A	**0A**	(00)
0000198B	**0B**	(D0) Total # of Bytes reserved for Rooms, Stairs down & Stairs up

Another very useful global setting is located in `Sector 0043`:

Game	Sector 0043	
Byte #	Bytes	
0000226F	**EF**	(00) (See Note)

The possible combinations for this byte are as follows:

- ➢00-70 = A Store is located on the Ground Floor
  - o 00-30 Player can use Fountains if dead (disabled), food and magic can heal
  - o 40-70 Player cannot us Fountains if dead, food and magic can not heal
- ➢80-F0 = No Store on Ground Floor
  - o 80-B0 Can use Fountain if dead (disabled), food and magic can heal
  - o C0-F0 Cannot use Fountain if dead, food and magic cannot heal

The hex value entered here determines ***both*** a Store's presence on the Ground floor ***and*** the possibility of healing if dead (disabled) by using Fountains, consuming food, or using Magical Items – In other words whether you are truly dead and not merely "disabled" as in ***Quest***. ***Pennies*** uses hex 80, ***Quest*** hex 00 and ***Sub12*** hex C0 for this byte. If a Player is truly dead, only healing purchased from a Store appears to "bring 'em back". Other useful combinations and effects might be determined here.

Miscellaneous Global Values

Game	Sector 0050		
Byte #	Bytes		
00002EC8	48-4C	(00)	(Any data written here immediately reverts back to "00")
00002ECD	4D	(02)	AV for "Hands". This value is written to Sector 002E (Player Stats) as Players' AV if no weapon is present, but is reset by module to "2" on use.
00002ECE	4E	(14)	Maximum Gold in Vaults (For First Floor, then multiples of floor)
00002ECF	4F	(1E)	??? Uncertain

Game	Sector 0043	
Byte #	Bytes	
000021E8	68-76	Monster selection by Exp Class per floor: (Easy = 68-6C; Med. = 6D-71; Hard = 72-76)

Game	Sector 0044	
Byte #	Bytes	
000022F6	76-79	# of Players & their order, i.e. 01, 02, 03, 04. This changes when you press O & change Order.
000022FA	7A-83	Max. # of Room Monsters by Floors: 1 Byte/ Floor (7A=Floor 1, 7B=2, 7C=3, etc.) (To this added # Players, difficulty, etc.) To set all Rooms on a floor with a set # of Monsters/ Room, use: FC=7, FB=6, FA=5, F9=4, F8=3, F7=2, F6=1, F5=0
00002304	84-8F	Dynamic workspace that represents room computations, contents, location of vault, chest, stairs, etc. Always '00' for halls
00002310	90-93	Initial Hit Points for the 4 Classes of Characters

Sector 0043, bytes 68-76 determines the number of monster types and their relative toughness used to populate each dungeon floor or level. Using ***Quest*** again as an example we have the following:

```
                Sector 0043
Byte #            0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
000021E0       6 00 00 00 71 02 00 4B 02 06 05 03 02 01 04 04 03  ...q..K.........
000021F0       7 03 03 02 02 03 04 06 00 20 59 4F 55 20 46 4F 55  ........ YOU FOU
```

These are the hex values that will be used when a player selects Easiest, Medium, or Hardest for Game Difficulty. Each group of 5 hex numbers represent 5 of the groups of 4 monster types classified by the Experience points a Player receives for vanquishing them. For convenience I have reprinted this table on the next page with a visual example. For Easiest Game Choice this will translate to: 6 Monster types from the 1st Experience Class group (Experience of 10), 5 Monster types from the 2nd (Experience of 30), 3 monster types from the 3rd (Experience of 60), 2 monster types from the 4th (Experience of 100) and 1 monster type from the 5th (Experience of 150). The net result of this sequence is that the majority of your opponents on the first floor will be selected from the two easiest (11 monsters total) categories available out of the five. Presuming the progression of monsters is from easiest to hardest, as was planned in Quest, this scheme works admirably. A choice of Medium would produce an almost even distribution (all 4's and 3's) and Hardest would choose 10 out of 17 monster types from the highest Experience groups. This progression follows through to successive floors along with the Experience assignment sequence.

You may have noticed that each selection of monster types by Game difficulty totals 17. This is not coincidence! What is further defined by these groupings is, "How many rooms do we wish to populate with Monsters?" In each example above, 17 rooms will have Monsters in them. Quest has 20 rooms per floor, 3 are reserved for potential Stores or Vaults, leaving 17 rooms to populate with *nasties*… A game developer should follow suit although the TOD program will ignore an excess number of monster types dropping from the upper end, or the toughest group, back. Poor planning = "less of a challenge".

Monsters by Experience Class - Example from Quest (wih Easiest Option)

#	Experience	Name	Lev-1	Lev-2	Lev-3	Lev-4	Lev-5	Lev-6	Lev-7	Lev-8	Lev-9	Lev-10	Lev-11	Lev-12
1	10 (6 Monsters from this group)	Goblin	1											
2		Kobold												
3		Rat												
4		Ooze												
5	30 (5 Monsters from this group)	Evil Mane		2										
6		Giant Rat												
7		Lizard												
8		Imp												
9	60 (3 Monsters from this group)	Orc			3									
10		Skeleton												
11		Wild Dog	Floor											
12		Spider												
13	100 (2 Monsters from this group)	Wolf				4								
14		Zombie												
15		Dark Slime		Floor										
16		Spider												
17	150 (1 Monster from this group)	Lemure					5							
18		Wight												
19		Dust Devil			Floor									
20		Gremlin	1											
21	210	Cursed One						6						
22		Metalloid												
23		Snake				Floor								
24		Vampire Bat		2										
25	280	Smogg							7					
26		Pixie												
27		Hobgoblin					Floor							
28		Ghost			3									
29	360	Metazoid								8				
30		Land Crab												
31		Whiplash						Floor						
32		Gnoll				4								
33	450	Troll									9			
34		Shambler												
35		Serpent							Floor					
36		Minotaur					5							
37	550	Ghoul										10		
38		Giant Wasp												
39		Ogre								Floor				
40		Devourer						6						
41	660	Vampire												
42		Scorpion												
43		Buzz Bomb									Floor			
44		Hill Giant							7					
45	780	Frost Giant												
46		Demon												
47		Tarantula										Floor		
48		Wyvern								8				
49	810	Demon King												
50		Dragon												
51		Elemental												
52		Pit Fiend									9			
53	950	Arch-Devil												
54		Land Shark												
55		Dragon Lord												
56		Will O'Wisp										10		

There is at least one additional use for these settings. Theoretically, there were no monster groups available to populate floors 11 and 12 in ***Sub12*** as the sequence simply ends at 56 Monsters. While more than one solution was likely, the one selected simply utilized fewer groups of monster types per floor. As each floor drops one Experience group of Monsters and adds the next, by drawing from only 3 groups (instead of 5) per floor the sequence was extended for another two floors. Thirteen floors is possible by utilizing only two monster groups and 14 floors with only one – thus only 4 monster types per floor (possibly boring). With our current state of knowledge, the theoretical limit for a ***Quest*** type game base is 14 floors. Beyond this only a ***Pennies & Prizes*** type database is readily available (with no monsters, etc.) Speaking for myself - four floors of that was plenty!

To continue a ***Tunnels of Doom*** game beyond 10 levels requires modification of half a dozen or so areas of the database. Some features will have to be significantly modified and others eliminated. That this is even possible is due to the lack of imposed database limits and error checking, by the ***Tunnels of Doom*** game program proper. Again, these are very likely due to memory constraints encountered during initial development. Using this presentation as guide, a game developer can modify ***Sub12*** toward this goal if that is desired.

Some additional Global Game Options:

Game **Byte #**	**Sector 0044** **Bytes**	
000022FA	7A-83	Max. # of Room Monsters by Floors: 1 Byte/ Floor (7A=Floor 1, 7B=Floor 2, 7C=Floor 3, etc.) (To this added # Players, difficulty, etc.) To set all Rooms on a floor with a set # of Monsters/ Room, use: FC=7, FB=6, FA=5, F9=4, F8=3, F7=2, F6=1, F5=0

```
              Sector 0044
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
000022F0    7 02 02 02 02 02 02 01 02 03 00 00 00 01 01 01 02 ................
00002300    8 02 03 03 03 0D 0F 00 00 00 00 00 00 00 00 00 00 ................
```

Sector 0044, bytes 7A-83 permits further refinement of monster distribution by determining (in part) the number of monsters distributed within rooms. ***Quest*** uses hex 00-03 as values. A hex value of 03 permits up to 3 additional monsters in a room, in addition to an amount determined by the number in the Party (4 max) and the game difficulty choice. Each byte above corresponds to one floor. The maximum number of monsters that can populate a room appears to be 7.

It is possible to set every room on a specified floor with a fixed number of monsters. This is achieved by placing the following hex values in the corresponding byte: **FC=7, FB=6, FA=5, F9=4, F8=3, F7=2, F6=1, F5=0** This technique should be used sparingly!

Miscellaneous Global Values

Game **Byte #**	**Sector 0050** **Bytes**		
00002EB8	38	(01)	Initial Value (x 10) for Experience assigned to Monsters in Monster List
00002EB9	39	(03)	Progression (x 10) of 1) Monster Experience Points, 2) Gold amounts in Rooms and Vaults
00002EBA	3A	(00)	

These last global values under consideration have far reaching effects, for they determine the initial value and progression of several attributes. Byte 38 sets the initial value for Experience gained for slaying monsters in the List of 56 monsters. Changing this will likely require modification of how a game database determines when to advance a Player to the next Level, the corresponding increase in Hit Points assigned as a result, the maximum range of Experience assigned to Monsters and the like.

Similarly, the progression value affects a number of game calculations, which must be anticipated for reasonable game play. The default value is hex 03 (x10) with a valid range of hex 00-03. Above hex 03 game behavior becomes erratic or freezes during the dungeon creation phase. Summary:

00 = all monsters are assigned the Experience value given by the hex digit in byte 38 (x10)
01 = the same formula used with 03, but monster values remain the same as the first floor, with an Experience range of 10-150 points. As the lowest Experience group is dropped and a new one added, their range remains 10-150.
02 = monsters are assigned the Experience value given in byte 38 (x10) for the first floor. On each successive floor, monster values increase by the initial amount. Thus 10-20-30-40, etc.
03 = byte 38 represents the initial value (x10). The Experience progression can be calculated as follows:

(to calculate the lowest monster experience value for each floor)

Initial value (byte 38)	10	10	10	10	10	10	10	10
Last Experience increase	10	20	30	40	50	60	70	80
Old Experience value	10	30	60	100	150	210	280	360
Next Experience Level	30	60	100	150	210	280	360	450

And so forth. Subsequent values on each floor follow the same sequence.

A fascinating, but heretofore little explored use of Tunnels of Doom is its potential use as a medium for creating Game scenarios or schemes. I am not referring to the random dungeons and arrangements typically generated by the game module, but predefined creations. Why leave so many game details up to random chance… and a frequently uncongenial game module? Why indeed?

This concept was suggested by the *Sample Quest* that is part of the Quest database. We have seen how a great many of the typical features, such as; vaults, stores, fountains, statues, chests and their contents can be customized, what about their placements? The short answer is yes, but it does require a bit of work. Before we get started however, we must be upfront about the shortfalls of using this technique.

- ➢ It ***is*** a lot of additional work. Remember, this will be in <u>addition</u> to what you will need to do for any new game database.
- ➢ Your dungeon design will not change and eventually will get boring. In anticipation, your overall design should accommodate both your scenario and a typical TOD random dungeon.
- ➢ Your scenario can <u>only</u> be played as designed with the *Continue Current Game* option. To a lesser degree, *Restock Old Dungeon* will also work, as far as dungeon design, but not for the contents.
- ➢ Your design will be very easily lost, or more correctly, over written. This will likely happen as soon as the first saved game unless the user is careful. Back in the diskette days, a write protect tab on the write notch served this purpose well. Many users had "flippy" diskettes (write notches on top and bottom of the disk) with one side write protected. An alternative is always (try to) remember to use a different filename when saving a game (works with emulators as well)!

It is best to break down your project into distinct steps, as was recommended for creating a game proper. You might proceed as follows: 1) Dungeon Design, 2) Contents & Monsters, 3) Party Stats.

For experimenting with Dungeon Design I would recommend using the Pennies database, modified to meet your specific number of rooms, fountains, etc. requirements as discussed in Chapter VIII. Using ***Pennies*** gives you the advantage of not needing maps to check out your various level designs, but more importantly, it will contain no monsters, which otherwise will invariably pop up when all you want to do is take a look at how your designs look. I have always used a modified Pennies database to test out new dungeon designs, hallway graphics, fountains, statues and the like. Utility software, such as ***DISKU***, make it easy to copy disk sectors from one disk drive to another, so once hallway and other graphics are to your liking you can copy them from *Pennies* to your actual "game disk in progress".

Dungeon Design (floor plans)

Designing dungeons is always experimental. While the game programmer can specify the exact location of a room, fountain, stairwell, etc., to create a design, the TOD program itself creates their connections based upon internal algorithms. It does not take long to get a knack for this and you will soon discover the top to bottom, left to right, design or bias of the algorithms involved. All floor placements as viewed on a TOD Map (which is displayed anytime you press <u>M</u> for Map during a game) have two bytes of data assigned to give its exact location on the floor. The breakdown for this is as follows:

First byte: The <u>first</u> hex digit is always a 0 (zero)
The <u>second</u> hex digit is a 0 = top of the map, or a 1 = bottom of the map
Note: For the first hex digit a 2 is also possible, but never used by the game.

Second byte: The <u>first</u> hex digit defines the <u>x</u> or horizontal coordinate on the map
If the <u>x</u> coordinate is an even (hex) number, then it refers to the left side of map
If <u>x</u> is an odd (hex) number, then the right side of the map
The *second* hex digit defines the <u>y</u> or vertical coordinate (the map column)
Note: Alternating even/ odd numbers for screen halves permits duplication of "y" numbers.

D	*E*	*F*	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*
		Y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	Y		
	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	
0	*0*	-																												*1*	0
	2																													3	
	4																													5	
	6																													7	
	8																													9	
	A																													B	
	C																													D	
	E																													F	
1	0																													1	1
	2																													3	
	4																													5	
	6																													7	
	8																													9	
	A																													B	
	C																													D	
	E																													F	
2	*0*																												-	*1*	*2*
	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	
												F	L	O	O	R	:		1												
				P	R	E	S	S		B	A	C	K		W	H	E	N		F	I	N	I	S	H	E	D				

Gray area represents colored Map Border. Dashed (**-**) areas are visible as Map, but not assigned Rooms, Hallways, etc. or saved with game.

Map display is 19 Rows x 28 columns. First and last Rows are always left blank leaving 17 Rows. First X-Y coordinate of Row one and Last of Row 17 are not used/ saved. The Italic, non-Bold columns/ rows exist, but are never used as locations.

Map Locations - Sequence of <u>4 Hex digits</u>: **1**$^{\text{st}}$= 0, **2**$^{\text{nd}}$ = 0 or 1 (0 = Top of Map, 1 = Bottom of Map. It is possible to manual insert a "2" as the second hex digit of a location, but no horizontal hallways will issue from there. The module never assigns a "2" as the second hex digit.) **3**$^{\text{rd}}$ = Row (X coordinate), even# = L, odd# = R side, **4**$^{\text{th}}$ = Column (Y coordinate).

Map Example

D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	A	B	C
	Y		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**		**Y**	
	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**X**	
0	*0*	-																												*1*	*0*
0	**2**																													**3**	**0**
	4																													**5**	
	6																													**7**	
	8																													**9**	
	A					00C3		00C5		00C7		00C9		00CB		00CD		00E0		00D1		00D3		00D5		00D7				**B**	
	C					x		x		x		x		↓		x		x		x		x		x		x				**D**	
	E		⊗		x		x		x		x		x		x				x		x		x		x			↑		**F**	
1	**0**		00E0		00E2		00E4		00E6		00E8		00EA		00EC		↓		00F0		00F2		00F4		00F6			00F9		**1**	**1**
	2																010E													**3**	
	4																													**5**	
	6																													**7**	
	8																									⊗				**9**	
	A																									0197				**B**	
	C																													**D**	
	E																													**F**	
2	*0*																												-	*1*	*2*
	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**X**	
												F	L	O	O	R	:		1												
				P	R	E	S	S		B	A	C	K		W	H	E	N		F	I	N	I	S	H	E	D				

First Floor Map Design from - *Halls of Lost… Moria!*

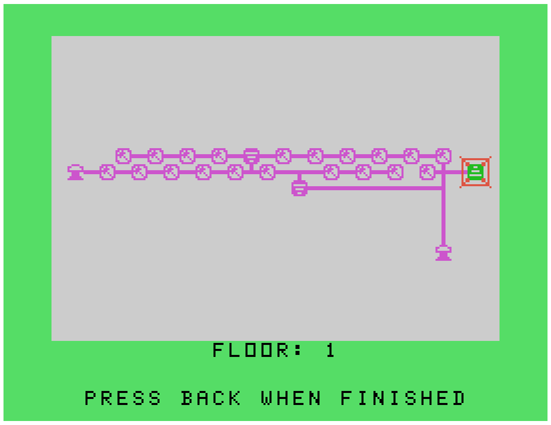
Example: First Hall Fountain location (left) is `hex 00E0`   The Second (right) is `hex 0197`

The Top/ Bottom and Left/ Right demarcations are indicated by thicker solid lines.
Above = 0, Below = 1, Left of = Even line number, Right of = odd line number, Columns = number directly above

Map Example

D	*E*	*F*	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0	1	2	3	4	5	6	7	8	9	*A*	*B*	*C*
	Y		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**		**Y**	
	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**X**	
0	*0*	-																								0017				*1*	*0*
0	**2**																								0036	X	0038			**3**	**0**
	4																							0055	X		X	0059		**5**	
	6																							x				x		**7**	
	8																									⊗				**9**	
	A																							x		00B7		x		**B**	
	C			00E1			00E4																	00D5	X		X	00D9		**D**	
	E			x			x																		00F6	X	00F8			**F**	
1	**0**																									0117				**1**	**1**
	2								x																	x				**3**	
	4								0146																	0157				**5**	
	6																													**7**	
	8																													**9**	
	A								x																					**B**	
	C								01C6																					**D**	
	E								x																					**F**	
2	*0*								01E6																				-	*1*	*2*
	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**X**	
												F	L	O	O	R	:		12												
				P	R	E	S	S		B	A	C	K		W	H	E	N		F	I	N	I	S	H	E	D				

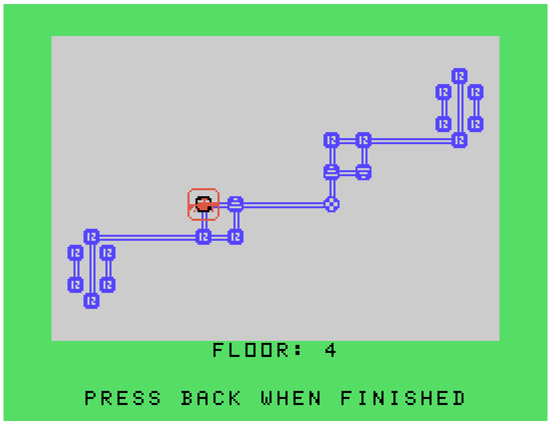
12th Floor Map Design from - *Sub-Level 12*



FLOOR: 1

PRESS BACK WHEN FINISHED

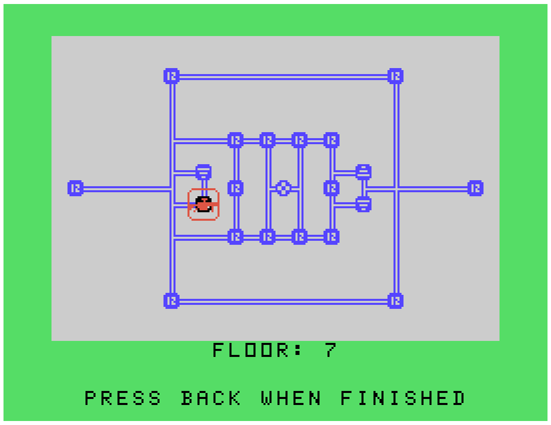
First Floor Map from *Halls*



FLOOR: 4

PRESS BACK WHEN FINISHED

Fourth Floor Map from *Sub12*



FLOOR: 7

PRESS BACK WHEN FINISHED

Seventh Floor Map from *Sub12*

Using the first floor layout of ***Halls*** as an example, will give you a good idea of how Map locations are defined within the TOD environment. Compare the Floor 1 Map screenshot on the above left with the previous labeled Floor 1 layout to see how this was done. Once locations have been plotted for a floor or level within a database, the only certain way to evaluate the outcome is to view the Map for that floor and see if that is what you had in mind. The algorithms used by TOD to create dungeon floors have a left to right "make a connection" bias as seen in the Floor 7 screenshot.

Dungeon Contents and Monsters

The location of a room is always the first byte, of ten, which are required to define each room. The other nine room bytes are used to store information on room contents. Stairs only require one byte for definition, to provide its location, which is also why you will never find monsters or item in stairways – there is no storage (memory) set aside for that option! Fountains require 1.5 bytes each, one byte for location, and a half byte – i.e. one hex digit, to inform the game of the fountain's status. Status defines from which of the four probability banks a Fountain commences divining its profundity (initial value). (See Chapter IV, the section on Fountains for details.) Two Fountains are able to share 3 bytes total for their information, but an odd numbered sequence of Fountains wastes one hex digit. So, the last fountain, if an odd number, takes 2 bytes. Fortunately, the master TI dungeon designer had a plan, so dungeons proceed in an orderly sequence during their creation.

The defined sequence for dungeon data storage is indicated in the Chapter VIII, **Designing Dungeons**: Rooms (regardless of their number) first; then Stairs Up, Stairs Down and lastly Fountains. This sequence is repeated for each floor or level of the dungeon. The 10 bytes of data storage in a TOD room are remarkably compact and efficient. Their expression as hex digits is summarized below:

Room	**bytes 1 & 2**	**bytes 3 & 4**	**bytes 5 & 6**	**bytes 7 & 8**	**bytes 9 &10**
	00 00	00 00	00 00	00 00	00 00

Bytes 1 & 2: Room location (coordinates) on Map
Byte 3: Flag for Store or Vault
Byte 4: Vault combination, or monster type and quantity
Byte 5: Flags for Living Statues, Fountains, status of Room & Chest
Byte 6: Amount of Gold in Room or Vault
Bytes 7-10: Flags, or indicators, for all other room items

The significance and determining factors of the Map location hex digits for rooms, stairways and fountains were described earlier in this chapter.

Byte 3:
This will be hex 01 if the room contains a Store, 02 if room contains a Vault, 00 if room contains neither. Note the consistency in the usage of flags for stores and vaults!

Byte 4:
If a Vault was flagged as present in the previous byte then the 2 hex digits expressed here represent the Vault's combination range. The first hex digit represents the number of digits in the combination, while the second hex represents its range in a, "from 1 to n" fashion, where n= the second hex digit.

If the previous byte did not indicate a Vault, then the <u>number</u> and <u>type</u> of monsters are expressed:

When the first hex digit is an <u>even</u> number, then <u>quantity</u> is expressed. If an odd number then both <u>quantity</u> and part of the monster's <u>number</u> are expressed. The second hex digit is always the monster's <u>number</u>, or part of it. The monster's <u>number</u> is not directly derived from the List of 56 Monster Types, but a number in hex, from 1-14 (1-20) from the monster pool used to stock a specific dungeon level. (You will recall that this list changes with every dungeon level, with the first group of four monsters being dropped from the pool and the next group of four being added.) In order to correctly identify a monster in a specific room, on a specific floor, you will need to consult that particular List!

Monster Hex digits: ("x" represents the range of hex numbers that follows)
00 = no monsters in the room
2x = one monster, from #1-F are in the room
    3x = one monster, from #10-14 are in the room
4x = two monsters, from #1-F are in the room
    5x = two monsters, from #10-14 are in the room
6x = three monsters, from #1-F
    7x = three monsters, from #10-14
8x = four monsters… etc.
Ax = five monsters… etc.
Cx = six monsters… etc.
Ex = seven monsters … etc.

The values of F0, F2, F3, F4 would represent <u>seven</u> monsters numbered (in hex): 10, 11, 12, 14 from that particular floor list. Using this scheme a maximum of 7 monsters – the TOD limit, can ever populate a room. If for any reason you opt to limit the number of monster types available for each floor pool, as e.g. in ***Sub12***, then this decision will be reflected. As Monsters are dispatched quantity goes to zero.

Byte 5: *(in brief, this byte represents Room objects a Party can interact with, but <u>not</u> take)*
The <u>first</u> hex digit may be flagged: 1= presence of Living Statue, {2, 6, A, E} = presence of a Room Fountain – the hex digits: 2, 6, A and E are the Room Fountain's equivalent of a Hall Fountain's 0, 1, 2, 3 - they designate which probability bank the Fountain will use (with E = best probabilities). A hex 0 means: neither a Living Statue nor Fountain is present.

The <u>second</u> hex digit, of byte 5, defines the <u>status</u> of Rooms and Chests

Room & Chest Status:
1 = unopened Chest
3 = unopened Chest with a trap
4 = ??? unused & unknown
8 = the Room has previously been entered and any Chest present opened
9 = the Room was entered, but the Chest was not opened
B = the Room was entered, but the Chest with a trap was not opened
This information is used to call forth the alternate color scheme to designate "rooms you have explored" on the Map. ***Quest*** uses the color Black to indicate "explored rooms".

Byte 6:
Indicates the amount of Gold in a Room or Vault (x 10).

The last four bytes express a unity of purpose, which is to indicate the presence of all other objects in the room: Magical items, Armor, Weapons, Maps and Quest items. These reference "Lists".

Byte 7: *(bytes 7-10 indicate Room objects a Party can interact with and take)*
The first hex digit can be a 0, 1, 2 or 3. 1 = a Map is present, 2 = Quest item is present, 3 = both a Map & Quest item are present, 0 = neither is present. The apparent purpose of these flags is to prioritize their placement in a room before it "gets full".

The second hex digit is used to define the graphics, types of items and their order of display in Room, Chest or Vault. The possibilities include:

1 = a Magical Item
3 = two Magical Items
4 = Armor
5 = Magical Item and Armor
7 = two Magical Items and Armor
8 = Weapon
9 = Magical Item and Weapon
B = 2 Magical Items and a Weapon
C = Armor and Weapon
D = Magical Item, Armor and Weapon

Note that these flags for room item are the same as those employed to seed empty rooms with items otherwise reserved for Stores or Vaults (only). There are a number of instances in ***Tunnels of Doom*** where a representation calls forth a result.

Byte 8:
Defines the first item (if any).

Byte 9:
Defines the second item (if any).

Byte 10:
Defines the third item (if any). Any Quest items are always indicated here.

A maximum of three items from the: Magical Item, Armor, Weapons, Map and Quest lists may be placed in a room. To distinguish these items from one another, a prefix hex digit (all even numbers) is joined to the item's number, which is its ordinal position (including blank fields) on its List. The prefixes are: 4 for Magical Items, 8 for Armor and Shields (range 81-8F), A for Weapons (range A1-B0), C for Maps (C0) and E for Quest items (range E1-E8). For example, in ***Quest,*** an A1 would indicate the presence of a *Dagger* in the room and A9 a *Sling*. Hex 81 would be *Leather Armor* and 8C the *Cloak of Hiding*. C0 would indicate the presence of a Map (it is the only item in the C category). E1 indicates the presence of the *King* in a room (Quest item #1) and E2 his *Rainbow Orb*. The specifics of a Quest item are always indicated in Room byte #10. The presence of one or more of the 40 Magical Items is flagged differently from the preceding objects in rooms.

The first Magical Item is simply entered in hex as 01 through 28 to correspond with the entries in the List of 40 Magical Items. Magical items are entered in the room bytes prior to any Armor or Weapons. A second Magical Item, if present, is prefixed by a 4. If the number of the Magical Item is >0F, (10 through 1F) then the 1 is added to the 4. So if the second Magical Item is #18 on the List, then it would appear in the Room byte as 58. Similarly, if the Magical Item was #24 in the List of 40 Magical Items, it would be represented by 64 in the room bytes. A maximum of two Magical Items can be found in Rooms, Chests or Vaults.

Some examples from Quest:

```
                Sector 003B
Byte #            0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001990       1 00 00 01 14 00 00 00 43 02 33 00 14 0C 83 AB 00 .......C.3......
```

Room location is 0043, there is a Vault in the room (byte 18) with 3 digits, ranging from 1 to 3 (byte 19). The Vault contains 14 or 200 in Gold (byte 1B). Other contents are: Armor & Weapon (byte 1C), which are Plate Mail (byte 1D) and a Crossbow (byte 1E).

```
                Sector 003C
Byte #            0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001A80       0 00 52 00 71 A1 1C 20 00 00 E2 01 12 00 6F 00 00 .R.q.. ......o..
```

Room location is 0052. There are three (71) monsters present, hex 11 (the 17th) in the Floor 2 monster list – "Cursed Ones" (byte 03). There is Room Fountain using probability bank #02 (first hex digit of byte 04) and an unopened Chest with no trap set (second hex digit of byte 04). The Chest contains 320 in Gold (byte 06) and the Rainbow Orb (byte 09).

```
                Sector 0042
Byte #            0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002150       D 00 A9 00 30 00 00 00 00 01 04 00 2B 01 3C 19 07 ...0.......+.<..
00002160       E A7 C0 00 6A 00 AB 00 00 00 00 00 00 01 C5 00 26 ...j...........&
```

Room location is 0104 (bytes D8 & D9). There is one monster present, hex 0C (12th) on the 10th Floor monster List (byte DB), a "Wyvem". The room contains an unopened Chest (byte DC) with 600 in Gold (byte DD), a Map, Magical item and Weapon (byte DE). The Magical Item is 07 (byte DF) or an "Aura of Warding" and the Weapon A7 (byte E0) a "Dwarven Ax". There is also a Floor Map C0 (byte E1).

Summary

Items are always placed in the room bytes in an ascending order of their prefixes: Magical Items, Armor, Weapons, Maps and Quest Items. If you wish to manually insert items into rooms it must be done in this order. Manual placement of items in rooms can occur regardless of whether Chests, Vaults or neither is present. ***Tunnels of Doom*** is rather forgiving of poor game database manipulations.

Limits for items are one each for Armor, Weapon, Maps, Quest Items and two for Magical Items. Total room limit for the above categories is three items. Quest items are always placed in the tenth (last) room byte, regardless if any other of the above items is present.

Dungeon Storage

The Dungeon contents of the last saved game begins in **Sector 003B**, **Byte 0C** and continues to **Sector 0043, Byte 67**. A game that is saved immediately after creation and prior to descent to the first Floor represents an excellent commencement point for a scenario.

The table on the next page provides the locations of all rooms, Stairways and Fountains, listed Floor by Floor as a reference to editing a TOD game database. However, these 2,140 bytes are completely fluid. If your game does not have the exact same specifications as Quest: 20 rooms, 2 stairways up & down and 2 fountains per floor; then this table will need to be recalculated to your specific Dungeon design for use as reference.

Remember, if you limit the number of floors in your dungeon to less than 10, or decrease the number of rooms from 20 per floor you can still use the following table. The TOD program will simply ignore any excess and unused allocated memory for floors and rooms. To verify that your computations and dungeon design are correct, set all the bytes from `Sector 003B, Byte 0C` through `Sector 0043, Byte 67` to `hex 00` before your dungeon is created. This will greatly simplify identification of any computation errors or glitches.

	Floor #1	Bytes	Floor #2	Bytes	Floor #3	Bytes	Floor #4	Bytes	Floor #5	Bytes
RM #1	Total Game Byte 0000198C	Sect. 003B 0C-15	Total Game Byte 00001A62	Sect. 003B E2-EB	Total Game Byte 00001B38	Sect. 003C B8-C1	Total Game Byte 00001C0E	Sect. 003D 8E-97	Total Game Byte 00001CE4	Sect. 003E 64-6D
RM #2	00001996	16-1F	00001A6C	EC-F5	00001B42	C2-CB	00001C18	98-A1	00001CEE	6E-77
RM #3	000019A0	20-29	00001A76	F6-FF	00001B4C	CC-D5	00001C22	A2-AB	00001CF8	78-81
RM #4	000019AA	2A-33	00001A80	Sect. 003C 00-09	00001B56	D6-DF	00001C2C	AC-B5	00001D02	82-8B
RM #5	000019B4	34-3D	00001A8A	0A-13	00001B60	E0-E9	00001C36	B6-BF	00001D0C	8C-95
RM #6	000019BE	3E-47	00001A94	14-1D	00001B6A	EA-F3	00001C40	C0-C9	00001D16	96-9F
RM #7	000019C8	48-51	00001A9E	1E-27	00001B74	F4-FD	00001C4A	CA-D3	00001D20	A0-A9
RM #8	000019D2	52-5B	00001AA8	28-31	00001B7E	FE-07	00001C54	D4-DD	00001D2A	AA-B3
RM #9	000019DC	5C-65	00001AB2	32-3B	00001B88	Sect. 003D 08-11	00001C5E	DE-E7	00001D34	B4-BD
RM #10	000019E6	66-6F	00001ABC	3C-45	00001B92	12-1B	00001C68	E8-F1	00001D3E	BE-C7
RM #11	000019F0	70-79	00001AC6	46-4F	00001B9C	1C-25	00001C72	F2-FB	00001D48	C8-D1
RM #12	000019FA	7A-83	00001AD0	50-59	00001BA6	26-2F	00001C7C	FC-05	00001D52	D2-DB
RM #13	00001A04	84-8D	00001ADA	5A-63	00001BB0	30-39	00001C86	Sect. 003E 06-0F	00001D5C	DC-E5
RM #14	00001A0E	8E-97	00001AE4	64-6D	00001BBA	3A-43	00001C90	10-19	00001D66	E6-EF
RM #15	00001A18	98-A1	00001AEE	6E-77	00001BC4	44-4D	00001C9A	1A-23	00001D70	F0-F9
RM #16	00001A22	A2-AB	00001AF8	78-81	00001BCE	4E-57	00001CA4	24-2D	00001D7A	FA-03
RM #17	00001A2C	AC-B5	00001B02	82-8B	00001BD8	58-61	00001CAE	2E-37	00001D84	Sect. 003F 04-0D
RM #18	00001A36	B6-BF	00001B0C	8C-95	00001BE2	62-6B	00001CB8	38-41	00001D8E	0E-17
RM #19	00001A40	C0-C9	00001B16	96-9F	00001BEC	6C-75	00001CC2	42-4B	00001D98	18-21
RM #20	00001A4A	CA-D3	00001B20	A0-A9	00001BF6	76-7F	00001CCC	4C-55	00001DA2	22-2B
Stairs Up	00001A54	D4-D7	00001B2A	AA-AD	00001C00	80-83	00001CD6	56-59	00001DAC	2C-2F
Stairs Down	00001A58	D8-DB	00001B2E	AE-B1	00001C04	84-87	00001CDA	5A-5D	00001DB0	30-33
Fountain #1	00001A5C	DC-DE	00001B32	B2-B4	00001C08	88-8A	00001CDE	5E-60	00001DB4	34-36
Fountain #2	00001A5F	DF-E1	00001B35	B5-B7	00001C0B	8B-8D	00001CE1	61-63	00001DB7	37-39
	Floor #6	Bytes	Floor #7	Bytes	Floor #8	Bytes	Floor #9	Bytes	Floor #10	Bytes
RM #1	00001DBA	3A-43	00001E90	10-19	00001F66	E6-EF	0000203C	BC-C5	00002112	92-9B
RM #2	00001DC4	44-4D	00001E9A	1A-23	00001F70	F0-F9	00002046	C6-CF	0000211C	9C-A5
RM #3	00001DCE	4E-57	00001EA4	24-2D	00001F7A	FA-03	00002050	D0-D9	00002126	A6-AF
RM #4	00001DD8	58-61	00001EAE	2E-37	00001F84	Sect. 0041 04-0D	0000205A	DA-E3	00002130	B0-B9
RM #5	00001DE2	62-6B	00001EB8	38-41	00001F8E	0E-17	00002064	E4-ED	0000213A	BA-C3
RM #6	00001DEC	6C-75	00001EC2	42-4B	00001F98	18-21	0000206E	EE-F7	00002144	C4-CD
RM #7	00001DF6	76-7F	00001ECC	4C-55	00001FA2	22-2B	00002078	F8-01	0000214E	CE-D7
RM #8	00001E00	80-89	00001ED6	56-5F	00001FAC	2C-35	00002082	Sect. 0042 02-0B	00002158	D8-E1
RM #9	00001E0A	8A-93	00001EE0	60-69	00001FB6	36-3F	0000208C	0C-15	00002162	E2-EB
RM #10	00001E14	94-9D	00001EEA	6A-73	00001FC0	40-49	00002096	16-1F	0000216C	EC-F5
RM #11	00001E1E	9E-A7	00001EF4	74-7D	00001FCA	4A-53	000020A0	20-29	00002176	F6-FF
RM #12	00001E28	A8-B1	00001EFE	7E-87	00001FD4	54-5D	000020AA	2A-33	00002180	Sect. 0043 00-09
RM #13	00001E32	B2-BB	00001F08	88-91	00001FDE	5E-67	000020B4	34-3D	0000218A	0A-13
RM #14	00001E3C	BC-C5	00001F12	92-9B	00001FE8	68-71	000020BE	3E-47	00002194	14-1D
RM #15	00001E46	C6-CF	00001F1C	9C-A5	00001FF2	72-7B	000020C8	48-51	0000219E	1E-27
RM #16	00001E50	D0-D9	00001F26	A6-AF	00001FFC	7C-85	000020D2	52-5B	000021A8	28-31
RM #17	00001E5A	DA-E3	00001F30	B0-B9	00002006	86-8F	000020DC	5C-65	000021B2	32-3B
RM #18	00001E64	E4-ED	00001F3A	BA-C3	00002010	90-99	000020E6	66-6F	000021BC	3C-45
RM #19	00001E6E	EE-F7	00001F44	C4-CD	0000201A	9A-A3	000020F0	70-79	000021C6	46-4F
RM #20	00001E78	F8-01	00001F4E	CE-D7	00002024	A4-AD	000020FA	7A-83	000021D0	50-59
Stairs Up	00001E82	Sect. 0040 02-05	00001F58	D8-DB	0000202E	AE-B1	00002104	84-87	000021DA	5A-5D
Stairs Down	00001E86	06-09	00001F5C	DC-DF	00002032	B2-B5	0000108	88-8B	000021DE	5E-61
Fountain #1	00001E8A	0A-0C	00001F60	E0-E2	00002036	B6-B8	0000210C	8C-8E	000021E2	62-64
Fountain #2	00001E8D	0D-0F	00001F63	E3-E5	00002039	B9-BB	0000210F	8F-91	000021E5	65-67

The Hex/ ASCII Image of the First Floor from Sample Quest

```
                Sector 003B
Byte #            0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001980       0 06 02 00 00 00 D6 00 C8 00 CC 00 D0 01 CE 00 00 ................
00001990       1 00 00 01 14 00 00 00 43 02 33 00 14 0C 83 AB 00 .......C.3......
000019A0       2 01 B7 02 33 00 14 05 0A 81 00 01 59 00 32 00 00 ...3.......Y.2..
000019B0       3 00 00 00 00 00 29 00 4F 00 14 00 00 00 00 00 D8 .....).O........
000019C0       4 00 4F 01 0C 10 C0 00 00 00 50 00 4B 00 0C 00 00 .O.......P.K....
000019D0       5 00 00 01 08 00 6B 00 00 00 00 00 00 00 75 00 6A .....k.......u.j
000019E0       6 10 0A 00 00 00 00 00 99 00 26 00 00 01 13 00 00 .........&......
000019F0       7 00 A8 00 25 01 06 01 08 00 00 01 18 00 65 00 0C ...%.........e..
00001A00       8 00 00 00 00 00 72 00 67 A0 00 00 00 00 00 01 AD .....r.g........
00001A10       9 00 64 00 00 01 0E 00 00 00 D6 00 41 00 06 00 00 .d.........A....
00001A20       A 00 00 01 12 00 41 00 00 01 0F 00 00 00 97 00 40 .....A.........@
00001A30       B 00 06 00 00 00 00 01 4B 00 22 00 0A 00 00 00 00 .......K."......
00001A40       C 01 D0 00 43 01 06 10 C0 00 00 00 00 00 00 00 00 ...C............
00001A50       D 00 00 00 00 01 73 00 00 01 43 01 A3 01 8C 03 01 .....s...C......
00001A60       E 2E 02 01 70 00 00 00 00 01 03 00 00 01 14 02 34 ...p...........4
00001A70       F 00 14 28 A3 00 E1 01 C5 02 34 00 28 18 AB C0 00 ..(......4.(....
```

The image above represents what ***Quest*** players encounter when they select Continue Current Game to walk through the introductory *Sample Quest* described on pages 14-18 of the ***Tunnels of Doom*** Command Module booklet. This scenario is intended as a training tool to introduce new players to the ***Tunnels of Doom*** world.

This image is that of Floor #1 and the first two rooms of Floor #2. The underlined groups of 10 bytes each are the room locations and their contents. Note that the last room consists of all "00" – it does not actually exist! The T.I ***Sample Quest*** actually only has 19 rooms on Floor #1. This occasionally happens and is usually corrected by selecting the option to create a new dungeon (giving the program a second chance, assuming the rooms, stairs, etc. bytes were correctly entered).

Quest reserves 3 rooms per floor for potential Stores and Vaults. Any reserved rooms are always presented first in the saved database, regardless of their locations on the Map. The Vault combinations, or where the Vault combination would be if one were present are highlighted in gray.

Stairs are only allocated enough memory to store their location, which is why you never encounter monsters or find items therein! The first Floor only has one set of stairs up to take you back to the General store, the only location on the Ground Floor. From the first floor there are two sets of Stairs down. The Fountains have an additional byte to indicate the probability bank for initial use. Here, one is set to bank "03" and the second to bank "02".

This Sector also contains the first three rooms of Floor #2, those reserved for Stores and Vaults. Both the first and second floors have two Vaults and were seeded with `hex 01`, causing Magical Items 14 to be placed in one room and 03 to be placed in the room on floor #2. This appears to have happened accidentally, only occurring on floors 1 and 2.

The bytes that store the Map locations for rooms, fountains and stairs are used in generating the Floor Map that becomes visible when pressing M during a game. There is only sufficient memory to store one Map at a time, which is always that of your current floor – whether it is visible or not. The Map generation is part of what is occurring in the background, when you descend or ascend to a new floor, and are treated to the TOD theme song. You may have noticed how *quickly* you ascend from Floor #1 to the General Store. There is only one location to plot out on the Ground Floor!

If in your scenario you wish to start your Party or Character on a floor other than the ground floor (e.g. ***Sub12*** begins with the Party on the 7th Floor) this Map must be coaxed into conforming with the layout of the intended floor to prevent conflicts.

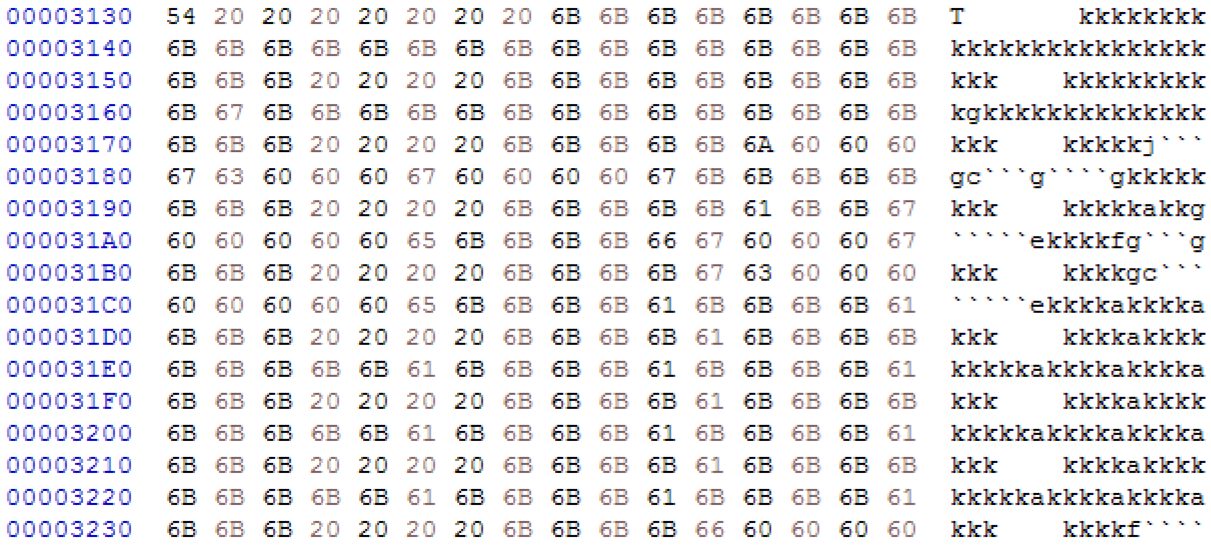
Map of Current Floor (including a Saved Game)

Game	Sector 0052		
Byte #	Bytes	Map Row	Border
00003130	B8-D2	1st Row of Map	D3-D6
00003150	D7-F2	2nd Row of Map	F3-F6
00003170	F7-12	3rd Row of Map	13-16
	Sector 0053		
00003190	17-32	4th Row of Map	33-36
000031B0	37-52	5th Row of Map	53-56
000031D0	57-72	6th Row of Map	73-76
000031F0	77-92	7th Row of Map	93-96
00003210	97-B2	8th Row of Map	B3-B6
00003230	B7-D2	9th Row of Map	D3-D6
00003250	D7-F2	10th Row of Map	F3-F6
00003270	F7-12	11th Row of Map	13-16 (Sector 0054)
	Sector 0054		
00003290	17-32	12th Row of Map	33-36
000032B0	37-52	13th Row of Map	53-56
000032D0	57-72	14th Row of Map	73-76
000032F0	77-92	15th Row of Map	93-96
00003310	97-B2	16th Row of Map	B3-B6
00003330	B7-D1	17th Row of Map	

The *character-codes* used to create each Map row are from the **Common Graphics Bank**.

Codes: 60-6F for explored sections of a Map
70-7F for unexplored sections of a Map

Note: The Map always represents the current Floor on which the Party is located. The Saved Map totals 17 Rows and reads from Top to Bottom, Left to Right. The first and last rows of a Saved Map are 27 bytes or characters wide, the other 15 Rows are 28 bytes. The Space Character '20' is used to provide a Border 2 bytes or characters wide, to the left and right of each row (except before the first & end of the last row). Thus each Row totals 32 Characters.



```
00003130  54 20 20 20 20 20 20 20 6B 6B 6B 6B 6B 6B 6B 6B  T       kkkkkkkk
00003140  6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B  kkkkkkkkkkkkkkkk
00003150  6B 6B 6B 20 20 20 20 6B 6B 6B 6B 6B 6B 6B 6B 6B  kkk    kkkkkkkkk
00003160  6B 67 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B  kgkkkkkkkkkkkkkk
00003170  6B 6B 6B 20 20 20 20 6B 6B 6B 6B 6B 6A 60 60 60  kkk    kkkkkj```
00003180  67 63 60 60 60 67 60 60 60 60 67 6B 6B 6B 6B 6B  gc```g````gkkkkk
00003190  6B 6B 6B 20 20 20 20 6B 6B 6B 6B 6B 61 6B 6B 67  kkk    kkkkkakkg
000031A0  60 60 60 60 60 65 6B 6B 6B 6B 66 67 60 60 60 67  `````ekkkkfg```g
000031B0  6B 6B 6B 20 20 20 20 6B 6B 6B 6B 67 63 60 60 60  kkk    kkkkgc```
000031C0  60 60 60 60 60 65 6B 6B 6B 6B 61 6B 6B 6B 6B 61  `````ekkkkakkkka
000031D0  6B 6B 6B 20 20 20 20 6B 6B 6B 6B 61 6B 6B 6B 6B  kkk    kkkkakkkk
000031E0  6B 6B 6B 6B 6B 61 6B 6B 6B 6B 61 6B 6B 6B 6B 61  kkkkkakkkkakkkka
000031F0  6B 6B 6B 20 20 20 20 6B 6B 6B 6B 61 6B 6B 6B 6B  kkk    kkkkakkkk
00003200  6B 6B 6B 6B 6B 61 6B 6B 6B 6B 61 6B 6B 6B 6B 61  kkkkkakkkkakkkka
00003210  6B 6B 6B 20 20 20 20 6B 6B 6B 6B 61 6B 6B 6B 6B  kkk    kkkkakkkk
00003220  6B 6B 6B 6B 6B 61 6B 6B 6B 6B 61 6B 6B 6B 6B 61  kkkkkakkkkakkkka
00003230  6B 6B 6B 20 20 20 20 6B 6B 6B 6B 66 60 60 60 60  kkk    kkkkf````
```

Part of a saved Floor Map from Quest

The exclusive use of Hex 60-6F character-codes indicate that the entire Floor was explored and/ or the Floor Map was Found

The Map is always present upon entering a Floor. It's creation is part of what is going on in the background when you Descend or Ascend a Floor, but is typically invisible with both Foreground & Background Colors being set to Gray. (Codes 70 - 7F) As a Floor is explored the `hex 70` series of *character-codes* are replaced with the `hex 60` series and thus become 'visible'. When a Map is found the module changes the Map color to Blue on Gray.

Party Stats

A saved game scenario presents the designer with much greater flexibility in assignment of Character and Party options than a new game. Virtually every Character attribute, for each Party Member may be individually adjusted. A full complement of Weapons and Armor may be assigned and up to 10 Magical Items provided. Party attributes can be modified as well, such as; gold, rations, maps on hand, etc.

The following two Sectors reflect the VDP memory of a saved game. This information is grouped into settings attributed to individual Party members and to those that affect the Party as a whole.

Several of the settings that deal with the Party as a whole can be modified in the same manner as the *Global Game Settings for New and Current Games*. This is particularly true for the various counter settings used to determine duration in paces or key presses.

In addition to the Sectors that determine Player, Party and Dungeon attributes of a Saved Game (or Scenario) the graphic representation for your Party will be those defined in **Sector 0026**.

The final portion of a Saved Game preserves the characteristics and hex settings of any Monsters that might have been present when the game was last saved. However this area of memory, the ***Monster Combat Queue*** may also reflect the last group of Monsters encountered and not those, in fact, currently present. The only way to determine which is which is by examining bytes 40-46 of Sector 002F to verify if all the hex values there are set to “00”. These bytes represent the individual Monster Hit Points and if all values here are “00” - they be all dead and none are present. This Queue is generally only completely updated when a new group of Monsters is encountered.

Player (Characters) Saved Game Parameters (Stats for: "Continue Current Game" Option)

	Game	Sector 002E						
	Byte #	Bytes		Bytes		Bytes		Bytes
Player Name #1	00000C80	00-0E	#2 00000CBC	3C-4A	#3 00000CF8	78-86	#4 00000D34	B4-C2
Hit Points	00000C8F	0F	00000CCB	4B	00000D07	87	00000D43	C3
# Wounds	00000C90	10	00000CCC	4C	00000D08	88	00000D44	C4
Armor Type	00000C90	11	00000CCD	4D	00000D09	89	00000D45	C5
Armor Protection	00000C92	12	00000CCE	4E	00000D0A	8A	00000D46	C6
Shield Type	00000C93	13	00000CCF	4F	00000D0B	8B	00000D47	C7
Shield Protection	00000C94	14	00000CD0	50	00000D0C	8C	00000D48	C8
Weapon #1 Type	00000C95	15	00000CD1	51	00000D0D	8D	00000D49	C9
Weapon AV	00000C96	16	00000CD2	52	00000D0E	8E	00000D4A	CA
#Ammo if ranged	00000C97	17	00000CD3	53	00000D0F	8F	00000D4B	CB
Weapon #2 Type	00000C98	18	00000CD4	54	00000D10	90	00000D4C	CC
Weapon AV	00000C99	19	00000CD5	55	00000D11	91	00000D4D	CD
#Ammo if ranged	00000C9A	1A	00000CD6	56	00000D12	92	00000D4E	CE
Armor Bonus Points	00000C9B	1B	00000CD7	57	00000D13	93	00000D4F	CF
Weapon Bonus	00000C9C	1C	00000CD8	58	00000D14	94	00000D50	D0
Players Luck	00000C1D	1D	00000CD9	59	00000D15	95	00000D51	D1
Experience	00000C1E	1E-1F	00000CDA	5A-5B	00000D16	96-97	00000D52	D2-D3
Last Exp. Gain	00000C20	20-21	00000CDC	5C-5D	00000D18	98-99	00000D54	D4-D5
Player's Level	00000C22	22	00000CDE	5E	00000D1A	9A	00000D56	D6
Player's Class	00000C23	23	00000CDF	5F	00000D1B	9B	00000D57	D7
???	00000C24	24	00000CE0	60	00000D1C	9C	00000D58	D8
Player's Abilities	00000C25	25	00000CE1	61	00000D1D	9D	00000D59	D9
Last (room) Location	00000C26	26-27	00000CE2	62-63	00000D1E	9E-9F	00000D5A	DA-DB
Magical Item #1	00000C28	28	00000CE4	64	00000D20	A0	00000D5C	DC
#Remaining Uses	00000C29	29	00000CE5	65	00000D21	A1	00000D5D	DD
Magical Item #2	00000C2A	2A	00000CE6	66	00000D22	A2	00000D5E	DE
#Remaining Uses	00000C2B	2B	00000CE7	67	00000D23	A3	00000D5F	DF
Magical Item #3	00000C2C	2C	00000CE8	68	00000D24	A4	00000D60	E0
#Remaining Uses	00000C2D	2D	00000CE9	69	00000D25	A5	00000D61	E1
Magical Item #4	00000C2E	2E	00000CEA	6A	00000D26	A6	00000D62	E2
#Remaining Uses	00000C2F	2F	00000CEB	6B	00000D27	A7	00000D63	E3
Magical Item #5	00000C30	30	00000CEC	6C	00000D28	A8	00000D64	E4
#Remaining Uses	00000C31	31	00000CED	6D	00000D29	A9	00000D65	E5
Magical Item #6	00000C32	32	00000CEE	6E	00000D2A	AA	00000D66	E6
#Remaining Uses	00000C33	33	00000CEF	6F	00000D2B	AB	00000D67	E7
Magical Item #7	00000C34	34	00000CF0	70	00000D2C	AC	00000D68	E8
#Remaining Uses	00000C35	35	00000CF1	71	00000D2D	AD	00000D69	E9
Magical Item #8	00000C36	36	00000CF2	72	00000D2E	AE	00000D6A	EA
#Remaining Uses	00000C37	37	00000CF3	73	00000D2F	AF	00000D6B	EB
Magical Item #9	00000C38	38	00000CF4	74	00000D30	B0	00000D6C	EC
#Remaining Uses	00000C39	39	00000CF5	75	00000D31	B1	00000D6D	ED
Magical Item #10	00000C3A	3A	00000CF6	76	00000D32	B2	00000D6E	EE
#Remaining Uses	00000C3B	3B	00000CF7	77	00000D33	B3	00000D6F	EF

Note: Armor, Shield and Weapon 'types' are their hex placement number from their respective lists. The Protection value is also from these lists plus any increased Protection obtained from Magical Items or Spells. A Weapon's AV is derived the same way. Bonuses are stored separately as they are inherent to a character, not their weapon or armor.

Ammo for ranged weapons purchased at stores appear to have a quantity limit of 120.

Experience is given 2 bytes, or 4 hex digits for expression. From Left to Right these hex digits represent the # of: 4096, 256, 16 & single - units. The total is multiplied by 10. Hex 1111 would = 43,690 Experience Points. Hex 333 = 8,190 Experience Points.

Class Name is one of the 4 possible Class or Status Names. This byte also defines the active weapon. Class #1=00 if Weapon #1 & 20 if Weapon #2 is active, Class #2=40 if #1 & 60 if #2, Class #3=80 if #1 & A0 if #2, Class #4=C0 if #1 & E0 if #2 is active. Active may='Hands'.

Bytes 24, 60, 9C & D8 are never populated in a saved game. It is possible that a character's graphic color was intended to be stored here, or more likely the Class Information was to extend over 2 bytes. The 1st byte indicating the Class Name and the 2nd to indicate the active weapon.

Magical Item numbers are from list of 40 Magical Items. If written in reverse notation they are unknown to the owner.

Party Saved Game Parameters

Game	Sector 002E		
Byte #	Bytes		
00000D70	F0-F7	(00)	Blank
00000D78	F8	(01)	Party location last time game was saved: 00=Top of Screen, 01=Bottom of screen
00000D79	F9	(73)	Party location last time game was saved. X-Y coordinates
00000D7A	FA	(00)	Current Floor
00000D7B	FB	(02)	Party's Speed baseline, or moves/ turn. Last one must be a combat move.
00000D7C	FC-FD	(000A)	Amount of Party Gold. Bytes FC=# of 4096 & 256 units, FD =# of 16 & 1 units. Total x 10.
00000D7E	FE	(00)	Map Flag - # Maps you have
00000D7F	FF	(00)	Blank

Note: Bytes F8-F9 = same value as Sector 0050, Bytes 3B-3C for Party hallway locations.
Byte FE is always the lowest level Floor Map the Party carries. If this byte reads '08' then you automatically have maps 1-8.

Party Saved Game Parameters (continued)

Game	Sector 002F		
Byte #	Bytes		
00000D80	00	#Quest Items found	A running total of the Quest Items found. Start = 00[1]
00000D81	01	#Quest Items remaining or destroyed	Identifies # of Quest Items at start of game[2]
00000D82	02-03	Time left for Quest Item #1	(Time remaining hex numbers are multiplied by lowest
00000D84	04-05	Time left for Quest Item #2	dungeon level.)
00000D86	06-07	Time left for Quest Item #3	
00000D88	08-09	Time left for Quest Item #4	
00000D8A	0A-0B	Time left for Quest Item #5	
00000D8C	0C-0D	Time left for Quest Item #6	
00000D8E	0E-0F	Time left for Quest Item #7	
00000D90	10-11	Time left for Quest Item #8	
00000D92	12	(03) Wandering Monster Probability	(baseline is obtained from 003A, byte DF)
00000D93	13	(14) Amount of Rations remaining	(Decrease by 1 for each Party member, including disabled
00000D94	14	(00)	& dead each time paces set in byte 17 is completed.)
00000D95	15	(00)	
00000D96	16	(00)	
00000D97	17	(02) Ration Consumption Interval	(baseline is obtained from 003A, byte DD)
00000D98	18	(00)	
00000D99	19	(02) Party Healing Interval	(baseline is obtained from 003A, byte E0)
00000D9A	1A	(00)	
00000D9B	1B	(00)	
00000D9C	1C	(00)	
00000D9D	1D	(00)	
00000D9E	1E	(00) Counter - ↑/↓ Combat Speed[3]	
00000D9F	1F	(04) Default - ↑/↓ Combat Speed[4]	
00000DA0	20	(00) Counter - ↑/↓ Wandering Monster Probability[3]	
00000DA1	21	(04) Default - Wandering Monster Probability[4]	
00000DA2	22	(00) Counter - ↑/↓ Consumption Interval[3]	
00000DA3	23	(04) Default - Consumption Interval change[4]	
00000DA4	24	(00) Counter - Healing Interval[3]	
00000DA5	25	(04) Default - Healing Interval change[4]	

[1] The number of each Quest Item is totaled in this byte when found. See numbers in parenthesis below. See example at end of this chapter.

[2] Initial value = # of Quest Items present: 1=01 (01), 2=03 (02), 3=07 (04), 4=0F (08), 5=1F (10), 6=3F (20), 7=7F (40), 8=FF (80). The hex number in parenthesis is the incremental difference, which doubles with each additional Quest Item to a max value of FF or 128. This incremental difference is added to byte 00 each time the corresponding item is found. Byte 01 then decrements using reverse notation (+1)

[3] Paces/ key presses remaining (x 10) for change from baseline.

[4] Change from baseline, used when the duration is not expressed in a spell or trap. Measured in paces/ key presses, e.g. 04=40 key presses.

Monster Combat Queue

Game Byte #	Sector 002F Bytes		
00000DA6	**26-31**	Monster's Name	Up to 12 characters.
00000DB2	**32**	Level	
00000DB3	**33**	DV	
00000DB4	**34**	AV	If value is in reverse notation, then a ranged attack.
00000DB5	**35**	Damage (Max Damage)	
00000DB6	**36**	Special Attack Chance %	
00000DB7	**37**	Special Attack Type (from list)	If value is in reverse notation, then a ranged attack.
00000DB8	**38**	Special Attack Damage	
00000DB9	**39**	Monster Sound Table Pointer	See Note.
00000DBA	**3A**	Monster Graphic Pointer	See Note.
00000DBB	**3B**	Negotiation	A number from 0 - 3, or 0-75%
00000DBC	**3C**	Mobility	A number from 1 - 4, or 25-100%
00000DBD	**3D**	Magical Resistance	0-A, or 0-100%
00000DBE	**3E**	Last hex # of monster attributes ???	Monster's Luck? Or nothing at all?
00000DBF	**3F**	Monster Speed	
00000DC0	**40**	HP Monster #1	Hit Points = (6 x Monster Level) x Random
00000DC1	**41**	HP Monster #2	
00000DC2	**42**	HP Monster #3	
00000DC3	**43**	HP Monster #4	
00000DC4	**44**	HP Monster #5	
00000DC5	**45**	HP Monster #6	
00000DC6	**46**	HP Monster #7	
00000DC7	**47**		
00000DC8	**48-49**	Monster #1 Location	See Room grid chart.
00000DCA	**4A-4B**	Monster #2 Location	
00000DCC	**4C-4D**	Monster #3 Location	
00000DCE	**4E-4F**	Monster #4 Location	
00000DD0	**50-51**	Monster #5 Location	
00000DD2	**52-53**	Monster #6 Location	
00000DD4	**54-55**	Monster #7 Location	
00000DD6	**56-57**		

Other Game Settings

00000DD8	**58**	(05)	
00000DD9	**59**	(0A)	
00000DDA	**5A**	(0F)	
00000DDB	**5B**	(03)	
00000DDC	**5C**	(0E) Lowering ↓ Party Combat Probability	Possibly baseline Party Luck
00000DDD	**5D**	(0A) Lowering ↑Party Combat Prob.	Possibly baseline Monster Luck
00000DDE	**5E**	(00)	
00000DDF	**5F**	(00)	
00000DE0	**60**	(1E) % Probability of Hearing Monster	When pressing L for Listening at door.
00000DE1	**61**	(28) % of not getting wounds at Vaults	When guessing incorrectly at Vault combination.
00000DE2	**62-71**	Word "Combination" - 16 bytes	

Note: These values are transferred from the List of 56 Monster Types as necessary.

Monster's sound and graphic pointers are in reversed order in the Combat Queue (compared to list) presumably because you can hear them before, when pressing 'L', or as you enter the room, before you see them.

Example of Quest Items Computation:

Sector 002F, **Byte #:**	00 Found	01 Remaining	Increment (doubles)
1 - Quest Item	00	01	01
2 - Quest Items	00	03	02
3 - Quest Items	00	07	04
4 - Quest items	00	0F	08
5 - Quest Items	00	1F	10
6 - Quest Items	00	3F	20
7 - Quest Items	00	7F	40
8 - Quest Items	00	FF	80

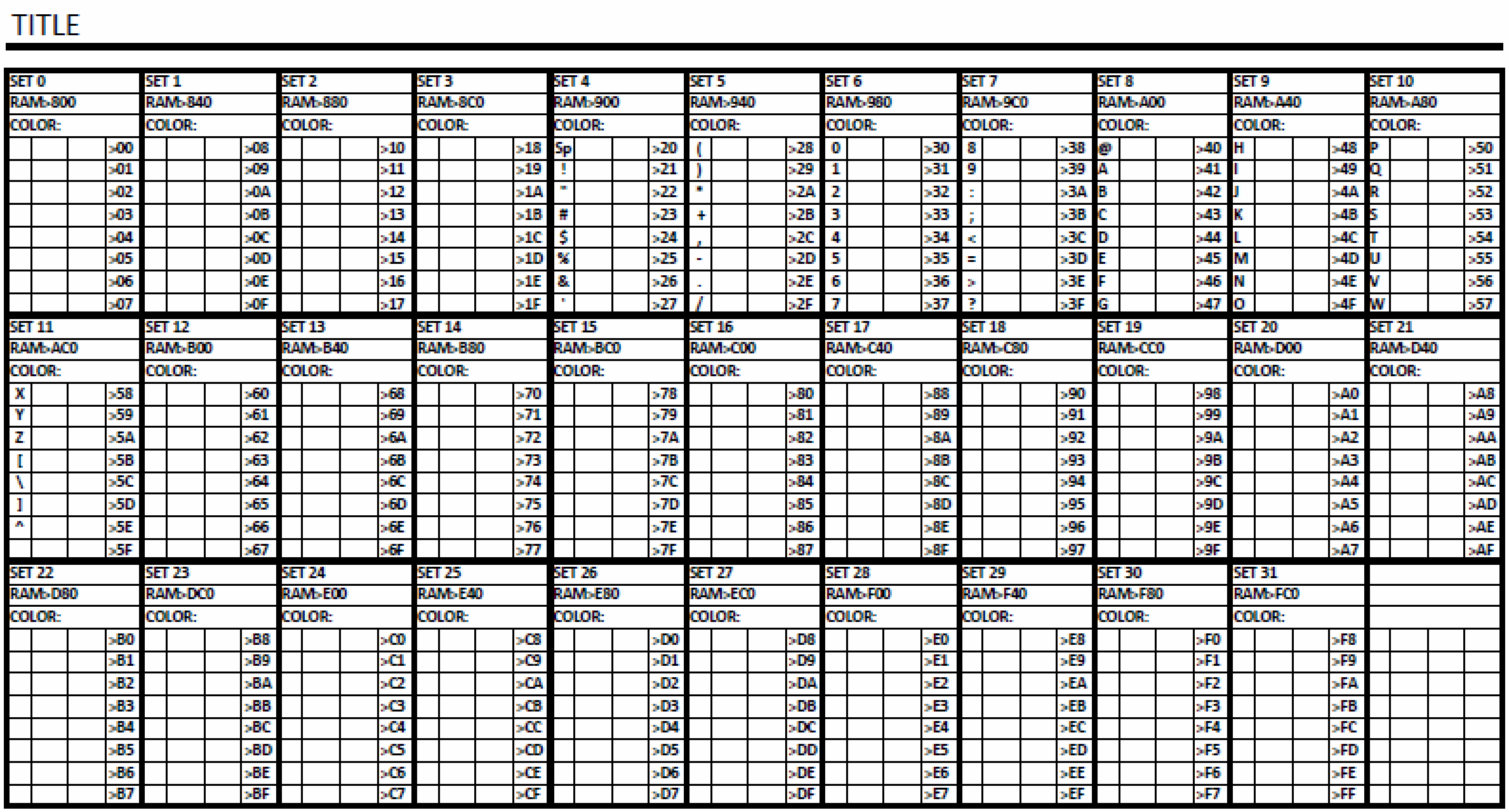
Initial Value for Remaining - (those yet to be found) is determined by the number of Quest Items.

If Quest Items were found in the order that they appear in the Quest List these values would change as follows - presuming 8 Quest Items:

Byte #:	00 Found	01 Remaining	Increment (doubles)
Initial value:	00	FF	00
First Found	01	FE	01
Second	03	FC	02
Third	07	F8	04
Fourth	0F	F0	08
Fifth	1F	E0	10
Sixth	3F	C0	20
Seventh	7F	80	40
Eighth	FF	00	80

The Remaining values are in the familiar reverse notation (+1) to indicate “yet to be found”. The final value is always **FF 00** when the last Item is found regardless of the number of Quest Items. As Quest Items are found their increment value is totaled in Byte 00 to let TOD know which ones have been found. If Items 2, 5 and 7 were found the value in **Byte 00** would be: 2+10+40 = **52** in hex and the value in **Byte 01** would decrease commensurately in reverse notation. If Items 1, 2 and 7 were found the values would be **43** and **BC**. As each increment number is only used once this system works quite well!

Appendix I - VDP Memory Character Descriptions



TITLE

SET 0		SET 1		SET 2		SET 3		SET 4		SET 5		SET 6		SET 7		SET 8		SET 9		SET 10	
RAM>800		RAM>840		RAM>880		RAM>8C0		RAM>900		RAM>940		RAM>980		RAM>9C0		RAM>A00		RAM>A40		RAM>A80	
COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:	
	>00		>08		>10		>18	Sp	>20	(	>28	0	>30	8	>38	@	>40	H	>48	P	>50
	>01		>09		>11		>19	!	>21	)	>29	1	>31	9	>39	A	>41	I	>49	Q	>51
	>02		>0A		>12		>1A	"	>22	*	>2A	2	>32	:	>3A	B	>42	J	>4A	R	>52
	>03		>0B		>13		>1B	#	>23	+	>2B	3	>33	;	>3B	C	>43	K	>4B	S	>53
	>04		>0C		>14		>1C	$	>24	,	>2C	4	>34	<	>3C	D	>44	L	>4C	T	>54
	>05		>0D		>15		>1D	%	>25	-	>2D	5	>35	=	>3D	E	>45	M	>4D	U	>55
	>06		>0E		>16		>1E	&	>26	.	>2E	6	>36	>	>3E	F	>46	N	>4E	V	>56
	>07		>0F		>17		>1F	'	>27	/	>2F	7	>37	?	>3F	G	>47	O	>4F	W	>57

SET 11		SET 12		SET 13		SET 14		SET 15		SET 16		SET 17		SET 18		SET 19		SET 20		SET 21	
RAM>AC0		RAM>B00		RAM>B40		RAM>B80		RAM>BC0		RAM>C00		RAM>C40		RAM>C80		RAM>CC0		RAM>D00		RAM>D40	
COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:	
X	>58		>60		>68		>70		>78		>80		>88		>90		>98		>A0		>A8
Y	>59		>61		>69		>71		>79		>81		>89		>91		>99		>A1		>A9
Z	>5A		>62		>6A		>72		>7A		>82		>8A		>92		>9A		>A2		>AA
[	>5B		>63		>6B		>73		>7B		>83		>8B		>93		>9B		>A3		>AB
\	>5C		>64		>6C		>74		>7C		>84		>8C		>94		>9C		>A4		>AC
]	>5D		>65		>6D		>75		>7D		>85		>8D		>95		>9D		>A5		>AD
^	>5E		>66		>6E		>76		>7E		>86		>8E		>96		>9E		>A6		>AE
	>5F		>67		>6F		>77		>7F		>87		>8F		>97		>9F		>A7		>AF

SET 22		SET 23		SET 24		SET 25		SET 26		SET 27		SET 28		SET 29		SET 30		SET 31			
RAM>D80		RAM>DC0		RAM>E00		RAM>E40		RAM>E80		RAM>EC0		RAM>F00		RAM>F40		RAM>F80		RAM>FC0			
COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:		COLOR:			
	>B0		>B8		>C0		>C8		>D0		>D8		>E0		>E8		>F0		>F8		
	>B1		>B9		>C1		>C9		>D1		>D9		>E1		>E9		>F1		>F9		
	>B2		>BA		>C2		>CA		>D2		>DA		>E2		>EA		>F2		>FA		
	>B3		>BB		>C3		>CB		>D3		>DB		>E3		>EB		>F3		>FB		
	>B4		>BC		>C4		>CC		>D4		>DC		>E4		>EC		>F4		>FC		
	>B5		>BD		>C5		>CD		>D5		>DD		>E5		>ED		>F5		>FD		
	>B6		>BE		>C6		>CE		>D6		>DE		>E6		>EE		>F6		>FE		
	>B7		>BF		>C7		>CF		>D7		>DF		>E7		>EF		>F7		>FF		

This printed Form represents the **Pattern Descriptor Table** stored in the TI99/4a VDP memory. It was used by TI in their designing of ***TOD*** and helps to visually understand the relationship of VDP memory and media storage as used in a TOD database. The table above was provided through the courtesy of Chris Schneider.

"The **Pattern Descriptor Table** contains descriptions of the 256 patterns or characters. By changing these descriptions, you can alter the appearance of the characters on screen. The description of each of the 256 patterns or characters takes eight bytes of information. The description of the subprogram CHAR in the User's Reference Guide discusses character definition."

See the **Editor/Assembler Manual**, page 329 for additional details.

Appendix II - Table of Negative Numbers or Reverse Notation

R.N. = Reverse Notation Value

R.N.	Hex	Dec		R.N.	Hex	Dec
FF	01	1		CD	33	51
FE	02	2		CC	34	52
FD	03	3		CB	35	51
FC	04	4		CA	36	54
FB	05	5		C9	37	55
FA	06	6		C8	38	56
F9	07	7		C7	39	57
F8	08	8		C6	3A	58
F7	09	9		C5	3B	59
F6	0A	10		C4	3C	60
F5	0B	11		C3	3D	61
F4	0C	12		C2	3E	62
F3	0D	13		C1	3F	63
F2	0E	14		C0	40	64
F1	0F	15		BF	41	65
F0	10	16		BE	42	66
EF	11	17		BD	43	67
EE	12	18		BC	44	68
ED	13	19		BB	45	69
EC	14	20		BA	46	70
EB	15	21		B9	47	71
EA	16	22		B8	48	72
E9	17	23		B7	49	73
E8	18	24		B6	4A	74
E7	19	25		B5	B	75
E6	1A	26		B4	4C	76
E5	1B	27		B3	4D	77
E4	1C	28		B2	4E	78
E3	1D	29		B1	4F	79
E2	1E	30		B0	50	80
E1	1F	31		AF	51	81
E0	20	32		AE	52	82
DF	21	33		AD	53	83
DE	22	34		AC	54	84
DD	23	35		AB	55	85
DC	24	36		AA	56	86
DB	25	37		A9	57	87
DA	26	38		A8	58	88
D9	27	39		A7	59	89
D8	28	40		A6	5A	90
D7	29	41		A5	5B	91
D6	2A	42		A4	5C	92
D5	2B	43		A3	5D	93
D4	2C	44		A2	5E	94
D3	2D	45		A1	5F	95
D2	2E	46		A0	60	96
D1	2F	47		9F	61	97
D0	30	48		9E	62	98
CF	31	49		9D	63	99
CE	32	50		9C	64	100

Appendix III - Quest - Hex/ ASCII Code

The following pages contain a ***Quest*** database as viewed with a PC Hex Editor. The file is in V9T9 format, but would be exactly the same in TIFILES format except for the *file header* information as shown in the first example below. The *file header* resides in the first 128 bytes of a V9T9 or TIFILES file and is not integral to a TOD database.

The second example shows the file header in TIFILES format. Fred’s excellent ***TIDir*** program will quickly convert from one format to the other.

```
[File header information] (V9T9)
------------------------
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000  51 55 45 53 54 20 20 20 20 20 00 00 01 00 00 33  QUEST     .....3
00000010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
[File header information] (TIFILES)
------------------------
00000000 07 54 49 46 49 4C 45 53 00 33 09 00 00 00 00 00 .TIFILES.3......
00000010 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 .S.S.S.S.S.S.S.S
00000020 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 .S.S.S.S.S.S.S.S
00000030 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 .S.S.S.S.S.S.S.S
00000040 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 .S.S.S.S.S.S.S.S
00000050 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 .S.S.S.S.S.S.S.S
00000060 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 .S.S.S.S.S.S.S.S
00000070 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 CA 53 .S.S.S.S.S.S.S.S
```

What follows on the next 17 pages is a TOD database referenced in both absolute bytes from the beginning of a TIFILES or V9T9 file, and in the sector/ bytes format seen on physical floppies and disk images. The intent here is to provide a convenient cross reference so the reader may easily convert any specific byte(s) between the two formats.

```
Game        Sector 0022
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000080  0  00 78 48 48 74 3A 35 32 31 30 30 78 48 48 78 00  .xHHt:52100xHHx.
00000090  1  00 1E 12 12 1E 0C 0C 8C 4C AC 5C 2E 12 12 1E 00  ........L.\.....
000000A0  2  00 7F 4F 48 78 60 63 7E 7E 63 60 78 48 4F 7F 00  ..OHx`c~~c`xHO..
000000B0  3  00 FE F2 12 1E 00 C0 40 40 C0 00 1E 12 F2 FE 00  .......@@.......
000000C0  4  00 7F 4F 48 78 60 60 7F 7F 00 00 78 48 4F 7F 00  ..OHx``....xHO..
000000D0  5  00 FE F2 12 1E 00 00 FE FE 06 06 1E 12 F2 FE 00  ................
000000E0  6  00 78 48 48 78 63 62 62 63 66 66 7E 4C 4C 78 00  .xHHxcbbcff~LLx.
000000F0  7  00 1E 12 12 1E C6 46 46 C6 66 66 7E 32 32 1E 00  ......FF.ff~22..
00000100  8  00 7F 7F 60 60 60 60 60 60 60 60 60 60 7F 7F 00  ...``````````...
00000110  9  00 FE FE 06 06 06 06 06 06 06 06 06 06 FE FE 00  ................
00000120  A  00 02 02 02 02 00 78 01 01 78 00 02 02 02 02 00  ......x..x......
00000130  B  00 40 40 40 40 00 1E 80 80 1E 00 40 40 40 40 00  .@@@@......@@@@.
00000140  C  00 00 00 01 03 02 02 03 01 00 00 00 61 33 1E 00  .............a3..
00000150  D  00 78 CC 86 00 00 00 80 C0 40 40 C0 80 00 00 00  .x.......@@.....
00000160  E  00 01 03 06 0C 08 09 0F 07 01 00 00 00 00 01 0F  ................
00000170  F  F0 80 00 00 00 00 80 E0 F0 90 10 30 60 C0 80 00  ...........0`...

Game        Sector 0023
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000180  0  00 10 30 60 40 40 60 31 19 0F 00 00 00 00 00 00  ..0`@@`1........
00000190  1  00 00 00 00 00 00 F0 98 8C 06 02 02 06 0C 08 00  ................
000001A0  2  00 00 00 00 80 81 81 C3 63 31 19 0F 00 00 00 00  ........c1......
000001B0  3  00 00 00 00 F0 98 8C C6 C3 81 81 01 00 00 00 00  ................
000001C0  4  00 00 00 00 00 00 00 01 00 00 00 00 00 00 00 00  ................
000001D0  5  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001E0  6  00 00 00 00 00 00 01 02 01 00 00 00 00 00 00 00  ................
000001F0  7  00 00 00 00 00 00 00 80 00 00 00 00 00 00 00 00  ................
00000200  8  00 00 00 00 00 00 02 01 02 00 00 00 00 00 00 00  ................
00000210  9  00 00 00 00 00 00 80 00 80 00 00 00 00 00 00 00  ................
00000220  A  00 00 01 01 09 04 02 01 39 02 04 08 00 00 00 00  ........9.......
00000230  B  00 00 00 00 10 20 40 9C 80 40 20 90 80 80 00 00  ..... @..@ .....
00000240  C  1C 22 4F 90 A6 A8 92 91 51 6A 20 25 52 48 27 18  ."O.....Qj %RH'.
00000250  D  00 F0 08 64 92 0A 49 95 94 54 14 A4 0A A2 64 18  ...d..I..T....d.
00000260  E  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000270  F  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

Game        Sector 0024
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000280  0  A0 A0 20 20 20 20 20 20 51 55 45 53 54 20 4F 46  ..      QUEST OF
00000290  1  20 54 48 45 20 4B 49 4E 47 20 20 20 20 20 A0 A0   THE KING     ..
000002A0  2  A0 A0 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F  ..______________
000002B0  3  5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F 5F A0 A0  ______________..
000002C0  4  A0 A0 20 20 20 55 50 20 54 4F 20 46 4F 55 52 20  ..   UP TO FOUR 
000002D0  5  50 4C 41 59 45 52 53 20 47 4F 20 20 20 20 A0 A0  PLAYERS GO    ..
000002E0  6  A0 A0 20 49 4E 54 4F 20 41 20 44 55 4E 47 45 4F  .. INTO A DUNGEO
000002F0  7  4E 20 42 45 4E 45 41 54 48 20 41 20 20 20 A0 A0  N BENEATH A   ..
00000300  8  A0 A0 20 52 55 49 4E 45 44 20 43 41 53 54 4C 45  .. RUINED CASTLE
00000310  9  2E 20 59 4F 55 52 20 54 41 53 4B 20 20 20 A0 A0  . YOUR TASK   ..
00000320  A  A0 A0 20 49 53 20 54 4F 20 52 45 53 43 55 45 20  .. IS TO RESCUE 
00000330  B  59 4F 55 52 20 4B 49 4E 47 2C 20 20 20 20 A0 A0  YOUR KING,    ..
00000340  C  A0 A0 20 43 41 50 54 55 52 45 44 20 42 59 20 54  .. CAPTURED BY T
00000350  D  48 45 20 4D 4F 4E 53 54 45 52 53 20 20 20 A0 A0  HE MONSTERS   ..
00000360  E  A0 A0 20 57 49 54 48 49 4E 20 54 48 45 20 44 55  .. WITHIN THE DU
00000370  F  4E 47 45 4F 4E 2C 20 42 45 46 4F 52 45 20 A0 A0  NGEON, BEFORE ..
```

```
Game        Sector 0025
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000380   0 A0 A0 20 48 45 20 49 53 20 45 58 45 43 55 54 45 .. HE IS EXECUTE
00000390   1 44 2E 20 59 4F 55 20 4D 55 53 54 20 20 20 A0 A0 D. YOU MUST   ..
000003A0   2 A0 A0 20 41 4C 53 4F 20 52 45 54 52 49 45 56 45 .. ALSO RETRIEVE
000003B0   3 20 54 48 45 20 4B 49 4E 47 27 53 20 20 20 A0 A0  THE KING'S   ..
000003C0   4 A0 A0 20 4F 52 42 20 4F 46 20 50 4F 57 45 52 2E .. ORB OF POWER.
000003D0   5 20 20 20 20 20 20 20 20 20 20 20 20 20 20 A0 A0               ..
000003E0   6 A0 A0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ..
000003F0   7 20 20 20 20 20 20 20 20 20 20 20 20 20 20 A0 A0               ..
00000400   8 F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000410   9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000420   A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000430   B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000440   C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000450   D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000460   E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000470   F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................

Game        Sector 0026
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000480   0 00 01 03 03 01 07 3B 79 79 7B 32 02 02 06 00 00 ......;yy{2.....
00000490   1 00 84 C4 C4 84 C4 EE A4 9C C0 40 40 40 60 00 00 ..........@@@`..
000004A0   2 00 01 03 03 09 05 3A 79 78 79 33 02 02 06 00 00 ......:yxy3.....
000004B0   3 00 80 C0 C0 80 C0 E0 60 A0 E0 C0 40 40 60 00 00 .......`...@@`..
000004C0   4 00 01 03 03 01 13 17 39 11 03 03 02 02 06 00 00 .......9........
000004D0   5 00 80 C0 C0 80 C0 E0 90 90 D0 C0 40 40 60 00 00 ...........@@`..
000004E0   6 00 01 03 03 11 2B 47 01 01 03 03 02 02 06 00 00 .....+G.........
000004F0   7 00 80 C0 C0 80 C0 F0 A0 80 C0 C0 40 40 60 00 00 ...........@@`..
00000500   8 00 29 83 13 31 13 17 09 01 03 03 03 02 06 00 00 .)..1...........
00000510   9 00 8A C4 CA 8A C4 E4 94 8C C4 C4 C4 44 64 00 00 ............Dd..
00000520   A 00 01 03 03 01 03 07 05 05 03 02 05 0A 16 00 00 ................
00000530   B 00 80 C4 CE 94 D8 E0 D0 B0 80 C0 C0 40 60 00 00 ............@`..
00000540   C 00 01 03 03 01 13 17 39 11 03 03 02 02 06 00 00 .......9........
00000550   D 00 80 C0 C0 80 C0 E0 90 90 D0 C0 40 40 60 00 00 ...........@@`..
00000560   E 00 01 03 03 11 2B 47 01 01 03 03 02 02 06 00 00 .....+G.........
00000570   F 00 80 C0 C0 80 C0 F0 A0 80 C0 C0 40 40 60 00 00 ...........@@`..

Game        Sector 0027
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000580   0 00 00 00 00 00 00 00 00 00 10 10 10 10 00 10 00 ................
00000590   1 00 28 28 00 00 00 00 00 00 28 7C 28 28 7C 28 00 .((......(|((|(.
000005A0   2 00 38 54 30 18 54 38 00 00 44 4C 18 30 64 44 00 .8T0.T8..DL.0dD.
000005B0   3 00 20 50 20 54 48 34 00 00 08 10 20 00 00 00 00 . P TH4.... ....
000005C0   4 00 08 10 10 10 10 08 00 00 20 10 10 10 10 20 00 ......... .... .
000005D0   5 00 44 28 7C 28 44 00 00 00 10 10 7C 10 10 00 00 .D(|(D.....|....
000005E0   6 00 00 00 00 30 10 20 00 00 00 00 7C 00 00 00 00 ....0. ....|....
000005F0   7 00 00 00 00 00 30 30 00 00 04 08 10 20 40 80 00 .....00..... @..
00000600   8 00 38 44 44 44 44 38 00 00 10 30 10 10 10 38 00 .8DDDD8...0...8.
00000610   9 00 38 44 08 10 20 7C 00 00 38 44 18 04 44 38 00 .8D.. |..8D..D8.
00000620   A 00 08 18 28 48 7C 08 00 00 78 40 78 04 44 38 00 ...(H|...x@x.D8.
00000630   B 00 38 40 78 44 44 38 00 00 7C 04 08 10 20 20 00 .8@xDD8..|...  .
00000640   C 00 38 44 38 44 44 38 00 00 38 44 44 3C 04 78 00 .8D8DD8..8DD<.x.
00000650   D 00 30 30 00 30 30 00 00 00 30 30 00 30 10 20 00 .00.00...00.0. .
00000660   E 00 10 38 54 10 10 10 10 00 00 00 7C 00 7C 00 00 ..8T.......|.|..
00000670   F 00 10 10 10 10 54 38 10 00 38 44 08 10 00 10 00 .....T8..8D.....
```

```
Game         Sector 0028
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000680  0 00 38 44 54 58 40 3C 00 00 38 44 44 7C 44 44 00 .8DTX@<..8DD|DD.
00000690  1 00 78 44 78 44 44 78 00 00 38 44 40 40 44 38 00 .xDxDDx..8D@@D8.
000006A0  2 00 78 44 44 44 44 78 00 00 7C 40 78 40 40 7C 00 .xDDDDx..|@x@@|.
000006B0  3 00 7C 40 78 40 40 40 00 00 38 44 40 4C 44 38 00 .|@x@@@..8D@LD8.
000006C0  4 00 44 44 7C 44 44 44 00 00 38 10 10 10 10 38 00 .DD|DDD..8....8.
000006D0  5 00 04 04 04 04 44 38 00 00 48 50 60 50 48 44 00 .....D8..HP`PHD.
000006E0  6 00 40 40 40 40 40 7C 00 00 44 6C 54 44 44 44 00 .@@@@@|..DlTDDD.
000006F0  7 00 44 64 54 54 4C 44 00 00 7C 44 44 44 44 7C 00 .DdTTLD..|DDDD|.
00000700  8 00 78 44 44 78 40 40 00 00 38 44 44 54 4C 3C 00 .xDDx@@..8DDTL<.
00000710  9 00 78 44 44 78 48 44 00 00 38 44 30 08 44 38 00 .xDDxHD..8D0.D8.
00000720  A 00 7C 10 10 10 10 10 00 00 44 44 44 44 44 38 00 .|.......DDDDD8.
00000730  B 00 44 44 44 44 28 10 00 00 44 44 44 54 54 28 00 .DDDD(...DDDTT(.
00000740  C 00 44 28 10 10 28 44 00 00 44 44 28 10 10 10 00 .D(..(D..DD(....
00000750  D 00 7C 08 10 20 40 7C 00 18 24 5A A1 A1 5A 24 18 .|.. @|..$Z..Z$.
00000760  E 00 00 10 08 7C 08 10 00 00 00 00 00 00 00 82 7C ....|..........|
00000770  F 00 38 38 38 38 38 82 7C 00 00 00 FF FF 00 00 00 .88888.|........

Game         Sector 0029
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000780  0 00 00 00 FF FF 00 00 00 18 18 18 18 18 18 18 18 ................
00000790  1 18 18 18 FF FF 18 18 18 18 18 18 FF FF 00 00 00 ................
000007A0  2 00 00 00 FF FF 18 18 18 18 18 18 F8 F8 18 18 18 ................
000007B0  3 18 18 18 1F 1F 18 18 18 FF 81 BD A5 A5 BD 81 FF ................
000007C0  4 7E E7 FF C3 FF 81 FF 7E 7E FF 81 FF C3 FF E7 7E ~......~~......~
000007D0  5 3C 66 DB BD BD DB 66 3C 00 00 00 00 00 00 00 00 <f....f<........
000007E0  6 83 83 83 83 83 83 83 83 C0 C0 C0 C0 C0 C0 C0 C0 ................
000007F0  7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000800  8 00 00 00 FF FF 00 00 00 18 18 18 18 18 18 18 18 ................
00000810  9 18 18 18 FF FF 18 18 18 18 18 18 FF FF 00 00 00 ................
00000820  A 00 00 00 FF FF 18 18 18 18 18 18 F8 F8 18 18 18 ................
00000830  B 18 18 18 1F 1F 18 18 18 FF 81 BD A5 A5 BD 81 FF ................
00000840  C 7E E7 FF C3 FF 81 FF 7E 7E FF 81 FF C3 FF E7 7E ~......~~......~
00000850  D 3C 66 DB BD BD DB 66 3C 00 00 00 00 00 00 00 00 <f....f<........
00000860  E 83 83 83 83 83 83 83 83 C0 C0 C0 C0 C0 C0 C0 C0 ................
00000870  F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................

Game         Sector 002A
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000880  0 FF FF FF FF FF FF FF FF 00 00 00 00 00 00 00 00 ................
00000890  1 FF FE FC F8 F0 E0 C0 80 FF 7F 3F 1F 0F 07 03 01 ..........?.....
000008A0  2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000008B0  3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000008C0  4 FF FF FF FF FF FF FF FF FF 3F 0F 03 00 00 00 00 .........?......
000008D0  5 FF FF FF FF FF 3F 0F 03 FF FC F0 C0 00 00 00 00 .....?..........
000008E0  6 FF FF FF FF FF FC F0 C0 00 00 00 00 00 00 00 00 ................
000008F0  7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000900  8 FF FF FF FF FF FF FF FF F2 F2 F2 F2 F2 F2 F2 F2 ................
00000910  9 4F 4F 4F 4F 4F 4F 4F 4F 00 00 00 00 00 00 00 00 OOOOOOOO........
00000920  A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000930  B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000940  C F0 FF FF C7 DB DB DB DB 00 00 F0 FF FF C7 DB DB ................
00000950  D 00 00 00 00 F0 FF FF C7 00 00 00 00 00 00 F0 FF ................
00000960  E 0F FF FF E3 DB DB DB DB 00 00 0F FF FF E3 DB DB ................
00000970  F 00 00 00 00 0F FF FF E3 00 00 00 00 00 00 0F FF ................
```

```
Game          Sector 002B
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000980   0 DB DB DB DB DB DB DB DB 00 00 00 00 00 00 00 00 ................
00000990   1 00 00 00 00 00 00 00 00 DB FF E7 C3 C3 E7 FF DB ................
000009A0   2 FF C7 DB DB DB DB DB DB FF E3 DB DB DB DB DB DB ................
000009B0   3 FF FF C3 DB DB DB DB DB DB DB DB DB DB C3 FF FF ................
000009C0   4 FF FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 ................
000009D0   5 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000009E0   6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
000009F0   7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000A00   8 FF FF FF FF FF FF FF FF FF FF BB 55 EF EF EF FF ...........U....
00000A10   9 00 03 0F 1F 3F 3F 7F 7F 00 C0 F0 F8 FC FC FE FE ....??..........
00000A20   A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0A000A30   B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000A40   C FF FF FF FF FF FF FF FF 80 80 C0 C0 E0 F0 FC FF ................
00000A50   D 01 01 03 03 07 0F 3F FF FF FF 00 00 00 00 00 00 ......?.........
00000A60   E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000A70   F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................

Game          Sector 002C
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000A80   0 DB D6 CC D8 F0 E0 C0 80 DB 6B 33 1B 0F 07 03 01 .........k3.....
00000A90   1 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000AA0   2 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000AB0   3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000AC0   4 00 00 00 00 00 00 00 00 0F 0F 0F 0F 00 00 00 00 ................
00000AD0   5 F0 F0 F0 F0 00 00 00 00 FF FF FF FF 00 00 00 00 ................
00000AE0   6 00 00 00 00 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F ................
00000AF0   7 F0 F0 F0 F0 0F 0F 0F 0F FF FF FF FF 0F 0F 0F 0F ................
00000B00   8 00 00 00 00 F0 F0 F0 F0 0F 0F 0F 0F F0 F0 F0 F0 ................
00000B10   9 F0 F0 F0 F0 F0 F0 F0 F0 FF FF FF FF F0 F0 F0 F0 ................
00000B20   A 00 00 00 00 FF FF FF FF 0F 0F 0F 0F FF FF FF FF ................
00000B30   B F0 F0 F0 F0 FF FF FF FF FF FF FF FF FF FF FF FF ................
00000B40   C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000B50   D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000B60   E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000B70   F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................

Game          Sector 002D
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000B80   0 00 49 70 05 69 5C C6 E0 18 3A 08 69 43 DA E0 23 .Ip.i\...:.iC..#
00000B90   1 3A 20 49 6B BC 10 E0 16 3A 05 69 70 BC 10 E0 19 : Ik....:.ip....
00000BA0   2 3A BE 0E 01 06 A0 44 C8 78 B0 1C 49 85 BD 1A 3C :.....D.x..I...<
00000BB0   3 06 80 1A 05 69 8D A4 B0 1C 78 06 A0 9A E2 49 9A ....i....x....I.
00000BC0   4 06 E0 0E 37 06 C0 42 06 A0 9A E2 BD 7E E0 26 3A ...7..B.....~.&:
00000BD0   5 06 A0 36 FC 06 A0 34 CA AF 1D 03 20 68 B9 06 C0 ..6...4.... h...
00000BE0   6 00 06 E0 0E 38 06 80 18 BF A3 9E 1E 6E 86 AF 1C ....8.......n...
00000BF0   7 FA 86 AF 1D 03 06 A0 76 65 4D CE E0 1D 3A 00 49 .......veM...:.I
0000OC00   8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000C10   9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000C20   A 0F 0F 01 1D 39 70 E3 07 0E 1C 00 01 03 00 00 00 ....9p..........
00000C30   B FF FF FF FF FF 3F BF 3F 07 77 E7 C0 8E 1C 38 70 .....?.?.w....8p
00000C40   C 00 03 04 08 11 22 14 09 16 10 09 06 08 13 0C 00 ....."..........
00000C50   D 00 80 78 8C 12 22 44 C8 10 A8 44 04 88 50 60 00 ..x.."D...D..P`.
00000C60   E 75 AF 32 40 4A 3D BA E0 23 3A 20 05 68 B9 D4 75 u.2@J=..#: .h..u
00000C70   F AF 32 3F 4A 90 06 A0 2C 4A 6C BC 12 13 AA 12 03 .2?J...,Jl......
```

```
Game         Sector 002E
Byte #       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000C80  0 45 52 49 43 20 53 45 41 42 4C 41 44 45 20 20 14 ERIC SEABLADE  .
00000C90  1 00 01 02 01 01 03 08 02 00 02 00 00 00 00 00 00 ................
00000CA0  2 00 00 00 00 00 0F 09 0D 00 00 00 00 00 00 00 00 ................
00000CB0  3 00 00 00 00 00 00 00 00 00 00 00 00 4D 41 55 56 ............MAUV
00000CC0  4 45 20 44 27 4F 52 4D 2D 4D 55 4C 12 00 01 02 00 E D'ORM-MUL.....
00000CD0  5 00 02 06 02 0A 06 28 00 00 00 00 00 00 00 00 80 ......(.........
00000CE0  6 00 2C 07 0D 00 00 00 00 00 00 00 00 00 00 00 00 .,..............
00000CF0  7 00 00 00 00 00 00 00 00 46 4F 52 45 53 54 41 4C ........FORESTAL
00000D00  8 4C 20 47 52 49 4D 4D 0F 00 00 00 00 00 01 04 02 L GRIMM.........
00000D10  9 09 02 14 00 00 00 00 00 00 00 00 40 00 10 09 0F ...........@....
00000D20  A 11 04 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000D30  B 00 00 00 00 4D 41 55 56 45 20 44 27 4F 52 4D 2D ....MAUVE D'ORM-
00000D40  C 4D 55 4C 30 02 01 01 00 00 03 08 0F 0B 08 4B 00 MUL0..........K.
00000D50  D 00 FF 02 04 00 56 05 A0 00 2C 07 0F 08 01 00 00 .....V..,.......
00000D60  E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000D70  F 00 00 00 00 00 00 00 00 01 73 00 02 00 0A 00 00 .........s......

Game         Sector 002F
Byte #       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000D80  0 00 03 00 18 00 1C 00 00 00 00 00 00 00 00 00 00 ................
00000D90  1 00 00 03 14 00 00 00 02 00 02 00 00 00 00 00 04 ................
00000DA0  2 00 04 00 04 00 04 00 00 00 00 00 00 00 00 00 00 ................
00000DB0  3 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000DC0  4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00000DD0  5 00 00 00 00 00 00 00 00 05 0A 0F 03 0E 0A 00 00 ................
00000DE0  6 1E 28 43 4F 4D 42 49 4E 41 54 49 4F 4E 20 20 20 .(COMBINATION
00000DF0  7 20 20 46 49 47 48 54 45 52 20 20 20 0A 0F 00 00   FIGHTER   ....
00000E00  8 00 00 00 00 00 00 00 00 57 49 5A 41 52 44 20 20 ........WIZARD
00000E10  9 20 20 05 10 00 00 00 00 00 00 11 04 00 00 52 4F   ............RO
00000E20  A 47 55 45 20 20 20 20 20 08 2C 00 00 00 00 00 00 GUE     .,......
00000E30  B 00 00 00 00 48 45 52 4F 20 20 20 20 20 20 0C 3F ....HERO      .?
00000E40  C 00 00 00 00 00 00 1A 06 00 00 00 01 03 03 01 07 ................
00000E50  D 3B 79 79 7B 32 02 02 06 00 00 00 84 C4 C4 84 C4 ;yy{2...........
00000E60  E EE A4 9C C0 40 40 40 60 00 00 00 01 03 03 09 05 ....@@@`........
00000E70  F 3A 79 78 79 33 02 02 06 00 00 00 80 C0 C0 80 C0 :yxy3...........

Game         Sector 0030
Byte #       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000E80  0 E0 60 A0 E0 C0 40 40 60 00 00 00 29 83 13 31 13 .`...@@`...)..1.
00000E90  1 17 09 01 03 03 03 02 06 00 00 00 8A C4 CA 8A C4 ................
00000EA0  2 E4 94 8C C4 C4 C4 44 64 00 00 00 01 03 03 01 03 ......Dd........
00000EB0  3 07 05 05 03 02 05 0A 16 00 00 00 80 C4 CE 94 D8 ................
00000EC0  4 E0 D0 B0 80 C0 C0 40 60 00 00 00 01 03 03 01 13 ......@`........
00000ED0  5 17 39 11 03 03 02 02 06 00 00 00 80 C0 C0 80 C0 .9..............
00000EE0  6 E0 90 90 D0 C0 40 40 60 00 00 00 01 03 03 11 2B .....@@`.......+
00000EF0  7 47 01 01 03 03 02 02 06 00 00 00 80 C0 C0 80 C0 G...............
00000F00  8 F0 A0 80 C0 C0 40 40 60 00 00 00 01 03 03 01 07 .....@@`........
00000F10  9 3B 79 79 7B 32 02 02 06 00 00 00 84 C4 C4 84 C4 ;yy{2...........
00000F20  A EE A4 9C C0 40 40 40 60 00 00 00 01 03 03 09 05 ....@@@`........
00000F30  B 3A 79 78 79 33 02 02 06 00 00 00 80 C0 C0 80 C0 :yxy3...........
00000F40  C E0 60 A0 E0 C0 40 40 60 00 00 47 4F 42 4C 49 4E .`...@@`..GOBLIN
00000F50  D 20 20 20 20 20 20 01 01 02 02 00 00 00 FE A0 22       .........."
00000F60  E 4B 4F 42 4F 4C 44 20 20 20 20 20 20 01 01 01 02 KOBOLD      ....
00000F70  F 00 00 00 F0 E0 22 52 41 54 20 20 20 20 20 20 20 ....."RAT
```

```
Game       Sector 0031
Byte #      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00000F80 0 20 20 01 00 01 02 00 00 00 E6 20 20 4F 4F 5A 45   ........  OOZE
00000F90 1 20 20 20 20 20 20 20 20 01 00 01 02 0A 09 01 07         ........
00000FA0 2 11 20 45 56 49 4C 20 4D 41 4E 45 20 20 20 01 00 . EVIL MANE   ..
00000FB0 3 02 02 0A FF 01 F0 A1 20 47 49 41 4E 54 20 52 41 ....... GIANT RA
00000FC0 4 54 20 20 20 02 01 01 03 00 00 00 E6 20 20 4C 49 T   ........  LI
00000FD0 5 5A 41 52 44 20 20 20 20 20 20 01 01 01 03 00 00 ZARD      ......
00000FE0 6 00 2A 70 20 49 4D 50 20 20 20 20 20 20 20 20 20 .*p IMP
00000FF0 7 01 02 02 02 0A 0C 0A 9F E2 22 4F 52 43 20 20 20 ........."ORC
00001000 8 20 20 20 20 20 20 02 02 02 02 00 00 00 F0 E0 22       ........."
00001010 9 53 4B 45 4C 45 54 4F 4E 20 20 20 20 01 02 02 03 SKELETON    ....
00001020 A 00 00 00 61 21 20 57 49 4C 44 20 44 4F 47 20 20 ...a! WILD DOG
00001030 B 20 20 02 01 01 04 00 00 00 2D 30 20 53 50 49 44   .......-0 SPID
00001040 C 45 52 20 20 20 20 20 20 01 02 03 02 00 00 00 45 ER      .......E
00001050 D 30 20 57 4F 4C 46 20 20 20 20 20 20 20 20 02 03 0 WOLF        ..
00001060 E 01 04 0A FC 01 2D 30 20 5A 4F 4D 42 49 45 20 20 .....-0 ZOMBIE
00001070 F 20 20 20 20 03 03 02 03 00 00 00 61 52 21 44 41     .......aR!DA

Game       Sector 0032
Byte #      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001080 0 52 4B 20 53 4C 49 4D 45 20 20 03 00 02 04 0F 05 RK SLIME  ......
00001090 1 01 07 11 20 53 50 49 44 45 52 20 20 20 20 20 20 ... SPIDER
000010A0 2 01 02 03 02 14 07 04 45 30 30 4C 45 4D 55 52 45 .......E00LEMURE
000010B0 3 20 20 20 20 20 20 03 02 03 04 00 00 00 6E 92 21       .......n.!
000010C0 4 57 49 47 48 54 20 20 20 20 20 20 20 02 03 03 04 WIGHT       ....
000010D0 5 00 00 00 1C A2 20 44 55 53 54 20 44 45 56 49 4C ..... DUST DEVIL
000010E0 6 20 20 02 02 FD 02 28 06 01 08 31 30 47 52 45 4D   ....(...10GREM
000010F0 7 4C 49 4E 20 20 20 20 20 01 04 FC 03 14 02 0A FF LIN     ........
00001100 8 A2 22 43 55 52 53 45 44 20 4F 4E 45 20 20 04 02 ."CURSED ONE  ..
00001110 9 03 06 00 00 00 6C 92 20 4D 45 54 41 4C 4C 4F 49 .....l. METALLOI
00001120 A 44 20 20 20 03 04 03 04 14 09 02 5A 20 20 53 4E D   .......Z  SN
00001130 B 41 4B 45 20 20 20 20 20 20 20 03 03 04 03 14 07 AKE       ......
00001140 C 08 B2 A0 20 56 41 4D 50 49 52 45 20 42 41 54 20 ... VAMPIRE BAT
00001150 D 02 04 03 03 0F 0A 01 A0 31 30 53 4D 4F 47 47 20 ........10SMOGG
00001160 E 20 20 20 20 20 20 03 06 03 03 32 0B 06 08 A1 20       ....2....
00001170 F 50 49 58 49 45 20 20 20 20 20 20 20 04 02 FC 04 PIXIE       ....

Game       Sector 0033
Byte #      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001180 0 05 FD 01 9F B3 22 48 4F 42 47 4F 42 4C 49 4E 20 ....."HOBGOBLIN
00001190 1 20 20 04 05 02 06 05 FC 01 7E E0 23 47 48 4F 53   .......~.#GHOS
000011A0 2 54 20 20 20 20 20 20 20 03 06 FB 04 14 0D 01 1B T       ........
000011B0 3 33 20 4D 45 54 41 5A 4F 49 44 20 20 20 20 04 04 3 METAZOID    ..
000011C0 4 04 04 1E 0F 04 8A E1 20 4C 41 4E 44 20 43 52 41 ....... LAND CRA
000011D0 5 42 20 20 20 02 07 03 08 00 00 00 CC 20 20 57 48 B   ........  WH
000011E0 6 49 50 4C 41 53 48 20 20 20 20 03 04 04 04 28 ED IPLASH    ....(.
000011F0 7 04 E6 32 30 47 4E 4F 4C 4C 20 20 20 20 20 20 20 ..20GNOLL
00001200 8 04 05 05 06 00 00 00 FB A0 34 54 52 4F 4C 4C 20 .........4TROLL
00001210 9 20 20 20 20 20 20 04 04 06 04 1E 0F 04 7E E1 25       .......~.%
00001220 A 53 48 41 4D 42 4C 45 52 20 20 20 20 0A 01 04 02 SHAMBLER    ....
00001230 B 0A 05 01 07 11 20 53 45 52 50 45 4E 54 20 20 20 ..... SERPENT
00001240 C 20 20 05 05 06 06 14 14 08 B2 20 30 4D 49 4E 4F   ........ 0MINO
00001250 D 54 41 55 52 20 20 20 20 06 06 05 06 00 00 00 8D TAUR    ........
00001260 E A1 24 47 48 4F 55 4C 20 20 20 20 20 20 20 05 04 .$GHOUL       ..
00001270 F FA 04 05 FD 01 9C 23 30 47 49 41 4E 54 20 57 41 ......#0GIANT WA
```

```
Game           Sector 0034
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001280     0 53 50 20 20 04 05 06 06 14 08 0C D4 30 30 4F 47 SP  ........00OG
00001290     1 52 45 20 20 20 20 20 20 20 20 05 06 04 08 05 FC RE        ......
000012A0     2 01 71 A0 25 44 45 56 4F 55 52 45 52 20 20 20 20 .q.%DEVOURER
000012B0     3 06 06 07 04 1E 09 03 5A 20 20 56 41 4D 50 49 52 .......Z  VAMPIR
000012C0     4 45 20 20 20 20 20 07 06 06 06 14 0A 04 13 A4 25 E     .........%
000012D0     5 53 43 4F 52 50 49 4F 4E 20 20 20 20 05 08 07 08 SCORPION    ....
000012E0     6 1E 08 0C CC 20 20 42 55 5A 5A 20 42 4F 4D 42 20 ....  BUZZ BOMB
000012F0     7 20 20 02 07 06 04 00 00 00 D4 30 40 48 49 4C 4C   ........0@HILL
00001300     8 20 47 49 41 4E 54 20 20 09 05 08 08 00 00 00 71  GIANT  .......q
00001310     9 F0 26 46 52 4F 53 54 20 47 49 41 4E 54 20 0A 06 .&FROST GIANT ..
00001320     A 07 0A 00 00 00 71 F0 28 44 45 4D 4F 4E 20 20 20 .....q.(DEMON
00001330     B 20 20 20 20 07 07 F9 06 14 FE 0F 83 B4 36 54 41     .........6TA
00001340     C 52 41 4E 54 55 4C 41 20 20 20 09 07 09 06 1E 08 RANTULA   ......
00001350     D 0C 45 30 20 57 59 56 45 52 4E 20 20 20 20 20 20 .E0 WYVERN
00001360     E 08 07 08 06 32 F2 08 39 72 24 44 45 4D 4F 4E 20 ....2..9r$DEMON
00001370     F 4B 49 4E 47 20 20 0A 09 F8 08 1E F3 03 83 B6 28 KING  .........(
```

```
Game           Sector 0035
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001380     0 44 52 41 47 4F 4E 20 20 20 20 20 20 0A 08 09 08 DRAGON      ....
00001390     1 1E F2 0C 39 F3 26 45 4C 45 4D 45 4E 54 41 4C 20 ...9.&ELEMENTAL
000013A0     2 20 20 05 0B F9 08 1E F0 04 98 75 30 50 49 54 20   ........u0PIT
000013B0     3 46 49 45 4E 44 20 20 20 0A 09 F9 0A 14 FF 0C 13 FIEND   ........
000013C0     4 B6 2A 41 52 43 48 2D 44 45 56 49 4C 20 20 0C 0A .*ARCH-DEVIL  ..
000013D0     5 F8 0C 0A EE 04 13 B7 2A 4C 41 4E 44 20 53 48 41 .......*LAND SHA
000013E0     6 52 4B 20 20 0C 0C 09 0C 28 11 14 5A 30 23 44 52 RK  ....(..Z0#DR
000013F0     7 41 47 4F 4E 20 4C 4F 52 44 20 0C 09 0A 0A 32 F2 AGON LORD ....2.
00001400     8 0C 39 F4 38 57 49 4C 4C 20 4F 27 20 57 49 53 50 .9.8WILL O' WISP
00001410     9 05 0C 04 06 0A FD 01 9B 35 50 54 4F 4E 47 55 45 ........5PTONGUE
00001420     A 20 4F 46 20 46 4C 41 4D 45 04 43 55 52 53 45 20  OF FLAME.CURSE
00001430     B 20 20 20 20 20 20 20 20 20 19 50 48 41 53 45 20          .PHASE
00001440     C 53 48 49 46 54 20 20 20 20 4C 52 41 47 45 20 20 SHIFT    LRAGE
00001450     D 20 20 20 20 20 20 20 20 20 50 43 4F 52 52 4F 53          PCORROS
00001460     E 49 4F 4E 20 20 20 20 20 20 09 53 41 4E 44 20 42 ION      .SAND B
00001470     F 4C 41 53 54 20 20 20 20 20 09 50 4F 49 53 4F 4E LAST     .POISON
```

```
Game           Sector 0036
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001480     0 20 42 49 54 45 20 20 20 20 04 50 4F 49 53 4F 4E  BITE    .POISON
00001490     1 20 53 54 49 4E 47 20 20 20 04 4D 45 54 41 4C 20  STING   .METAL
000014A0     2 45 41 54 49 4E 47 20 20 20 0D 42 4C 4F 4F 44 20 EATING   .BLOOD
000014B0     3 44 52 41 49 4E 20 20 20 20 01 53 55 46 46 4F 43 DRAIN    .SUFFOC
000014C0     4 41 54 49 4F 4E 20 20 20 20 04 50 52 41 4E 4B 20 ATION    .PRANK
000014D0     5 20 20 20 20 20 20 20 20 20 19 4C 49 46 45 20 44          .LIFE D
000014E0     6 52 41 49 4E 20 20 20 20 20 1D 42 52 45 41 54 48 RAIN     .BREATH
000014F0     7 45 53 20 46 49 52 45 20 20 04 52 45 47 45 4E 45 ES FIRE  .REGENE
00001500     8 52 41 54 49 4F 4E 20 20 20 70 45 41 52 54 48 51 RATION   pEARTHQ
00001510     9 55 41 4B 45 20 20 20 20 20 28 43 52 55 53 48 49 UAKE     (CRUSHI
00001520     A 4E 47 20 43 48 4F 4D 50 20 04 50 49 54 43 48 46 NG CHOMP .PITCHF
00001530     B 4F 52 4B 20 20 20 20 20 20 01 54 41 49 4C 20 57 ORK      .TAIL W
00001540     C 48 49 50 20 20 20 20 20 20 04 43 4F 4E 53 54 52 HIP      .CONSTR
00001550     D 49 43 54 49 4F 4E 20 20 20 04 00 00 00 01 06 07 ICTION   .......
00001560     E 18 33 1A 08 05 04 18 27 18 00 00 00 78 CC 22 96 .3.....'....x.".
00001570     F 4A CC 08 B4 44 28 90 A0 60 00 00 03 04 08 11 22 J...D(..`......"
Game          Sector 0037
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001580   0  14 09 16 10 09 06 08 13 0C 00 00 80 78 8C 12 22 .............x.."
00001590   1  44 C8 10 A8 44 04 88 50 60 00 00 00 01 02 02 01 D...D..P`.......
000015A0   2  07 0E 0E 1E 1D 3D 3A 72 76 00 00 00 80 40 40 80 .....=:rv....@@.
000015B0   3  E0 70 70 78 B8 BC 5C 4E 6E 00 00 00 01 02 22 71 .ppx..\Nn....."q
000015C0   4  17 18 04 0E 1E 3D 39 72 76 00 00 00 80 40 44 8E .....=9rv....@D.
000015D0   5  E8 18 20 70 78 BC 9C 4E 6E 00 00 00 00 00 00 00 .. px..Nn.......
000015E0   6  1F 3F 7F 7F 71 61 40 60 00 00 00 00 08 0C 1B 3E .?..qa@`.......>
000015F0   7  FC F0 E0 E0 C0 80 80 C0 00 00 00 00 00 00 00 03 ................
00001600   8  0F 1F 3F 7C 78 30 60 40 60 00 08 1C 3A 7F FC F0 ..?|x0`@`...:...
00001610   9  E0 E0 F0 FC 02 00 00 00 00 00 3C 56 AA 57 AB 5F ..........<V.W._
00001620   A  B1 E1 C3 03 03 C3 77 3E 1C 00 00 18 34 7E F8 EC ......w>....4~..
00001630   B  E0 F0 FA CC C6 C0 80 00 00 00 38 54 AD 55 AF 57 ..........8T.U.W
00001640   C  BF 67 C3 03 03 C3 77 3E 1C 00 00 C0 A1 F3 C6 4D .g....w>.......M
00001650   D  86 E3 F0 98 94 80 00 00 00 00 00 00 00 00 00 00 ................
00001660   E  30 4B 95 27 4A 93 24 48 48 00 00 00 00 00 00 00 0K.'J.$HH.......
00001670   F  0C D2 A9 E4 52 C9 24 12 12 00 00 00 01 23 55 4E ....R.$......#UN

Game          Sector 0038
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001680   0  46 0B 39 43 0F 13 63 04 18 00 00 00 80 C4 AA 72 F.9C..c........r
00001690   1  62 D0 9C C2 F0 C8 C6 20 18 00 00 01 01 01 03 07 b...... ........
000016A0   2  0F 39 7D 6F 65 66 33 78 48 00 00 80 80 80 C0 E0 .9}oef3xH.......
000016B0   3  F0 9C BE F6 A6 66 CC 1E 12 00 01 07 1F 39 7B FF .....f.......9{.
000016C0   4  ED CD C8 CA CF E7 60 70 38 28 80 E0 F8 9C DE FF ......`p8(......
000016D0   5  B7 B3 13 53 F3 E7 06 0E 1C 14 00 00 00 00 81 40 ...S...........@
000016E0   6  22 14 08 04 00 01 02 02 04 00 00 00 00 80 C0 80 "...............
000016F0   7  A0 90 8E 8E 8C 40 20 20 10 00 00 01 02 04 09 14 .....@  ........
00001700   8  02 00 00 00 00 01 02 02 04 00 80 00 00 80 C0 80 ................
00001710   9  00 E0 E0 C0 00 40 20 20 10 00 03 0F 1F 17 07 03 .....@  ........
00001720   A  1F 3F 77 67 6F 4F 0F 1C 1C 78 06 CF EF A6 86 0E .?wgoO...x......
00001730   B  EC FC 80 80 C0 C0 E0 E0 E0 78 30 73 7B 1D 0E 07 .........x0s{...
00001740   C  0B 0D 0C 09 01 03 07 07 07 1E C0 F0 F8 C0 F0 78 ...............x
00001750   D  B8 F0 C0 20 E0 F0 F8 38 38 1E 02 22 41 43 63 39 ... ...88.."ACc9
00001760   E  1F 07 03 01 01 02 04 08 30 00 40 44 82 C2 C6 9C ........0.@D....
00001770   F  F8 E0 C0 80 80 40 20 10 0C 00 00 8A DA 71 63 33 .....@ ......qc3

Game          Sector 0039
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001780   0  39 1F 0F 03 03 03 06 0C 18 70 04 51 44 91 CA C4 9........p.QD...
00001790   1  86 E6 FE C0 C0 C0 60 30 10 18 00 40 01 08 01 21 ......`0...@...!
000017A0   2  00 03 05 18 00 01 02 0C 00 00 00 04 20 C2 E0 20 ............ .. 
000017B0   3  C6 F8 E0 C0 C0 44 28 10 00 00 00 00 08 49 22 00 .....D(......I".
000017C0   4  CA 07 22 49 08 00 00 00 03 00 00 00 24 18 3C 24 .."I........$.<$
000017D0   5  18 FC 1A 1A 19 3D 44 84 06 00 00 00 00 20 72 9B .....=D...... r.
000017E0   6  8D 06 03 01 02 02 00 00 00 00 00 00 00 04 4E D9 ..............N.
000017F0   7  B1 60 C0 80 40 40 00 00 00 00 00 10 30 20 22 33 .`..@@......0 "3
00001800   8  1D 06 03 01 02 04 00 00 00 00 00 08 0C 04 44 CC ..............D.
00001810   9  B8 60 C0 80 40 20 00 00 00 00 00 01 03 07 07 03 .`..@ ..........
00001820   A  01 00 00 00 00 03 0E 1F 3F 00 00 00 80 C0 C0 80 ........?.......
00001830   B  80 80 C0 60 C0 80 00 C0 E0 00 00 00 01 03 03 01 ...`............
00001840   C  01 03 06 0C 18 0E 07 1F 3F 00 00 80 C0 E0 E0 C0 ........?.......
00001850   D  80 00 00 00 00 00 80 C0 E0 00 30 78 40 C0 80 80 ..........0x@...
00001860   E  C0 F3 7F 1F 0F 05 09 08 04 00 00 00 00 00 0E 38 ...............8
00001870   F  46 C0 F8 F8 F0 46 38 8E 40 00 00 1C 2E 70 A0 C0 F....F8.@....p..
```

```
Game          Sector 003A
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001880  0 E0 D0 BF 76 36 1F 09 19 11 18 00 00 02 04 08 3C ...v6..........<
00001890  1 62 C0 F0 E8 F8 C2 7C 18 84 C2 00 00 38 7C 0F 00 b.....|.....8|..
000018A0  2 3D 7F 7D 41 22 04 00 00 00 00 00 00 00 00 00 98 =.}A"...........
000018B0  3 B4 F8 80 80 40 80 00 00 00 00 60 70 38 1C 07 01 ....@.....`p8...
000018C0  4 01 0F 1D 1C 1D 1C 04 02 00 00 00 00 00 00 00 18 ................
000018D0  5 B4 FC 80 C0 20 90 00 00 00 00 01 07 18 20 24 05 .... ........ $.
000018E0  6 03 05 07 06 03 0F 1F 1B 31 F0 E0 38 04 04 22 A2 ........1..8..".
000018F0  7 C2 A2 E4 6C C8 F8 F8 D8 8C 0F 0E 10 20 24 15 03 ...l........ $..
00001900  8 05 07 06 E6 B3 1F 0F 03 05 0C 00 00 00 20 A0 C0 ............. ..
00001910  9 A8 E4 62 62 C4 F8 F0 D8 8C 07 00 00 00 50 22 27 ..bb.........P"'
00001920  A 32 1F 0F 0F 0F 0D 08 18 00 00 00 00 00 00 00 00 2...............
00001930  B 00 C0 F0 98 94 80 80 C0 00 00 00 00 01 03 03 01 ................
00001940  C 3F FF A7 A7 07 07 0E 0C 1C 00 00 00 80 C0 CA 8A ?...............
00001950  D FE F8 E0 E0 E0 E0 70 30 38 00 04 0A 00 02 02 03 ......p08.......
00001960  E 02 FC 0A 28 0A 14 02 02 01 07 03 02 14 01 03 0A ...(............
00001970  F 00 00 00 3C 02 02 03 00 00 00 00 01 06 0A 1E 06 ...<............
```

```
Game          Sector 003B
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001980  0 06 02 00 00 00 D6 00 C8 00 CC 00 D0 01 CE 00 00 ................
00001990  1 00 00 01 14 00 00 00 43 02 33 00 14 0C 83 AB 00 .......C.3......
000019A0  2 01 B7 02 33 00 14 05 0A 81 00 01 59 00 32 00 00 ...3.......Y.2..
000019B0  3 00 00 00 00 00 29 00 4F 00 14 00 00 00 00 00 D8 .....).O........
000019C0  4 00 4F 01 0C 10 C0 00 00 00 50 00 4B 00 0C 00 00 .O.......P.K....
000019D0  5 00 00 01 08 00 6B 00 00 00 00 00 00 00 75 00 6A .....k.......u.j
000019E0  6 10 0A 00 00 00 00 00 99 00 26 00 00 01 13 00 00 .........&......
000019F0  7 00 A8 00 25 01 06 01 08 00 00 01 18 00 65 00 0C ...%.........e..
00001A00  8 00 00 00 00 00 72 00 67 A0 00 00 00 00 00 01 AD .....r.g........
00001A10  9 00 64 00 00 01 0E 00 00 00 D6 00 41 00 06 00 00 .d.........A....
00001A20  A 00 00 01 12 00 41 00 00 01 0F 00 00 00 97 00 40 .....A.........@
00001A30  B 00 06 00 00 00 00 01 4B 00 22 00 0A 00 00 00 00 .......K."......
00001A40  C 01 D0 00 43 01 06 10 C0 00 00 00 00 00 00 00 00 ...C............
00001A50  D 00 00 00 00 01 73 00 00 01 43 01 A3 01 8C 03 01 .....s...C......
00001A60  E 2E 02 01 70 00 00 00 00 01 03 00 00 01 14 02 34 ...p...........4
00001A70  F 00 14 28 A3 00 E1 01 C5 02 34 00 28 18 AB C0 00 ..(......4.(....
```

```
Game          Sector 003C
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001A80  0 00 52 00 71 A1 1C 20 00 00 E2 01 12 00 6F 00 00 .R.q.. ......o..
00001A90  1 00 00 00 00 01 F4 00 4D 00 0A 00 00 00 00 00 48 .......M.......H
00001AA0  2 00 4B 00 05 00 00 00 00 01 91 00 28 01 10 05 03 .K.........(....
00001AB0  3 83 00 00 77 00 49 00 00 00 00 00 00 01 2D 00 65 ...w.I.......-.e
00001AC0  4 01 00 03 11 54 00 00 67 00 66 00 00 04 8A 00 00 ....T..g.f......
00001AD0  5 01 20 00 27 00 06 00 00 00 00 01 8C 00 67 10 0A . .'.........g..
00001AE0  6 00 00 00 00 00 83 00 64 01 0A 13 06 43 C0 01 CA .......d....C...
00001AF0  7 00 60 00 05 00 00 00 00 01 0B 00 21 00 09 00 00 .`.........!....
00001B00  8 00 00 01 77 00 63 00 00 00 00 00 00 00 73 00 23 ...w.c.......s.#
00001B10  9 00 0F 00 00 00 00 00 29 00 62 00 05 00 00 00 00 .......).b......
00001B20  A 00 4D 00 61 00 00 00 00 00 00 01 43 01 A3 00 00 .M.a.......C....
00001B30  B 00 00 00 44 03 01 37 02 01 E2 00 00 08 00 00 00 ...D..7.........
00001B40  C 00 00 01 25 00 00 08 00 08 A9 00 00 00 51 00 00 ...%.........Q..
00001B50  D 08 00 00 00 00 00 00 74 00 10 08 00 00 00 00 00 .......t........
00001B60  E 01 AA 00 0C 08 00 0C 83 A3 00 00 43 00 0D 08 00 ...........C....
00001B70  F 00 00 00 00 00 85 00 09 08 00 00 00 00 00 01 16 ................
```

```
Game        Sector 003D
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001B80  0  00 08 08 00 00 00 00 00 00 D8 00 08 08 00 00 00 ................
00001B90  1  00 00 00 E5 00 05 08 00 00 00 00 00 00 91 00 05 ................
00001BA0  2  18 00 00 00 00 00 01 A5 00 05 E8 00 00 00 00 00 ................
00001BB0  3  00 AA 00 04 08 00 00 00 00 00 4A 00 07 08 00 00 ..........J.....
00001BC0  4  00 00 00 00 00 A4 00 01 08 00 00 00 00 00 01 C8 ................
00001BD0  5  00 01 08 00 00 00 00 00 00 98 00 01 08 00 00 00 ................
00001BE0  6  00 00 00 56 00 00 08 00 00 00 00 00 00 AE 00 01 ...V............
00001BF0  7  08 00 00 00 00 00 00 32 00 00 08 00 00 00 00 00 .......2........
00001C00  8  00 D2 01 EE 01 6E 00 F3 01 D1 03 00 25 02 01 18 .....n......%...
00001C10  9  01 00 08 00 00 00 00 00 01 23 00 00 08 00 08 AA .........#......
00001C20  A  00 00 01 A1 00 00 08 00 04 83 00 00 00 43 00 13 .............C..
00001C30  B  08 00 00 00 00 00 01 C3 00 0C 08 00 00 00 00 00 ................
00001C40  C  00 D1 00 0C 08 00 00 00 00 00 01 E5 00 08 08 00 ................
00001C50  D  00 00 00 00 00 27 00 09 08 00 00 00 00 00 00 86 .....'..........
00001C60  E  00 0A 08 00 00 00 00 00 00 68 00 07 08 00 08 A2 .........h......
00001C70  F  00 00 00 39 00 07 08 00 00 00 00 00 00 37 00 06 ...9.........7..

Game        Sector 003E
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001C80  0  08 00 04 83 00 00 00 F7 00 07 08 00 00 00 00 00 ................
00001C90  1  00 58 00 04 08 00 00 00 00 00 00 95 00 00 08 00 .X..............
00001CA0  2  00 00 00 00 00 4A 00 01 18 00 00 00 00 00 00 CF .....J..........
00001CB0  3  00 00 08 00 00 00 00 00 00 B4 00 02 18 00 00 00 ................
00001CC0  4  00 00 00 4C 00 02 28 00 00 00 00 00 00 C1 00 00 ...L..(.........
00001CD0  5  08 00 00 00 00 00 01 6E 00 F3 00 A7 01 35 01 B0 .......n.....5..
00001CE0  6  03 00 20 02 00 D2 00 00 00 00 00 00 00 00 00 4D .. ............M
00001CF0  7  02 45 00 34 05 20 82 00 01 C9 00 00 08 00 00 00 .E.4. ..........
00001D00  8  00 00 01 21 00 93 00 00 01 11 00 00 01 49 00 6F ...!.........I.o
00001D10  9  01 24 01 04 00 00 01 A5 00 2F 00 23 00 00 00 00 .$......./.#....
00001D20  A  01 F3 00 2B 10 1C 00 00 00 00 00 B0 00 2B 01 1C ...+.........+..
00001D30  B  01 03 00 00 00 B4 00 88 11 0A 04 8A 00 00 00 75 ...............u
00001D40  C  00 A5 60 14 00 00 00 00 01 47 00 A6 01 28 11 1E ..`......G...(..
00001D50  D  C0 00 01 52 00 05 08 00 00 00 00 00 00 44 00 26 ...R.........D.&
00001D60  E  01 00 0D 01 8B AC 00 61 00 47 00 00 00 00 00 00 .......a.G......
00001D70  F  01 AD 00 02 08 00 00 00 00 00 01 00 00 41 00 00 .............A..

Game        Sector 003F
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001D80  0  08 AD 00 00 01 CF 00 02 08 00 00 00 00 00 01 0C ................
00001D90  1  00 A2 00 1E 00 00 00 00 01 2B 00 A1 E1 0A 08 AD .........+......
00001DA0  2  00 00 00 E4 00 60 00 08 00 00 00 00 00 A7 01 35 .....`.........5
00001DB0  3  01 31 00 28 01 25 03 00 C9 02 01 AE 00 00 00 00 .1.(.%..........
00001DC0  4  00 00 00 00 00 F3 02 46 00 77 11 09 C0 00 01 B8 .......F.w......
00001DD0  5  02 46 00 69 0D 0C 8A A3 01 B4 00 B2 00 00 00 00 .F.i............
00001DE0  6  00 00 01 D3 00 2C 00 00 00 00 00 00 00 59 00 AC .....,.......Y..
00001DF0  7  00 3C 00 00 00 00 00 F6 00 6B 00 28 00 00 00 00 .<.......k.(....
00001E00  8  01 50 00 89 00 2C 00 00 00 00 01 74 00 6A 03 28 .P...,.....t.j.(
00001E10  9  01 23 00 00 01 2F 00 25 11 00 0C 8B A2 00 00 CE .#.../.%........
00001E20  A  00 24 00 1C 00 00 00 00 01 ED 00 47 03 08 01 0F .$.........G....
00001E30  B  00 00 00 50 00 25 10 1C 00 00 00 00 00 21 00 86 ...P.%.......!..
00001E40  C  01 1E 01 17 00 00 01 60 00 40 A1 20 01 18 00 00 .......`.@. ....
00001E50  D  00 F9 00 20 00 00 01 0E 00 00 00 E4 00 41 60 08 ... .........A`.
00001E60  E  00 00 00 00 00 57 00 41 00 10 00 00 00 00 00 C8 .....W.A........
00001E70  F  00 43 01 10 14 8A C0 00 01 35 00 62 00 1B 00 00 .C.......5.b....
```

```
Game          Sector 0040
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001E80   0  00 00 01 31 00 28 01 90 00 F0 01 C0 03 00 EA 02 ...1.(..........
00001E90   1  01 63 00 00 00 00 00 00 00 00 00 AF 02 55 00 8C .c...........U..
00001EA0   2  00 00 00 00 00 C0 02 55 00 8C 13 03 4C C0 01 85 .......U....L...
00001EB0   3  00 B0 00 00 08 A9 00 00 01 98 00 2D 00 24 00 00 ...........-.$..
00001EC0   4  00 00 00 99 00 AF E1 34 04 85 00 00 00 64 00 4B .......4.....d.K
00001ED0   5  01 1E 10 C0 00 00 00 CA 00 2A 00 00 00 00 00 00 .........*......
00001EE0   6  00 8E 00 28 11 2D 01 1B 00 00 00 26 00 27 01 00 ...(.-.....&.'..
00001EF0   7  05 09 85 00 00 82 00 45 00 1B 00 00 00 00 00 D7 .......E........
00001F00   8  00 87 00 37 00 00 00 00 01 22 00 44 00 00 08 AB ...7.....".D....
00001F10   9  00 00 01 EA 00 47 03 00 03 1E 52 00 00 F8 00 A2 .....G....R.....
00001F20   A  00 24 00 00 00 00 01 C9 00 80 01 0B 01 26 00 00 .$...........&..
00001F30   B  01 E7 00 40 00 09 00 00 00 00 01 C1 00 20 A0 08 ...@......... ..
00001F40   C  00 00 00 00 01 67 00 63 00 0A 00 00 00 00 01 82 .....g.c........
00001F50   D  00 61 10 00 04 83 00 00 01 90 00 F0 00 40 01 93 .a...........@..
00001F60   E  00 C6 03 01 D9 02 00 D6 01 00 00 00 00 00 00 00 ................
00001F70   F  00 C1 02 56 00 A0 09 02 AB 00 01 64 02 56 00 A0 ...V.......d.V..
```

```
Game          Sector 0041
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00001F80   0  11 1F C0 00 01 2E 00 50 00 24 00 00 00 00 01 80 .......P.$......
00001F90   1  00 6F 00 00 00 00 00 00 01 12 00 8D 00 00 00 00 .o..............
00001FA0   2  00 00 01 59 00 69 01 00 07 04 59 86 00 8B 00 69 ...Y.i....Y....i
00001FB0   3  A0 24 00 00 00 00 00 B5 00 4B A0 00 00 00 00 00 .$.......K......
00001FC0   4  01 53 00 87 00 00 01 13 00 00 00 AE 00 27 00 00 .S...........'..
00001FD0   5  08 A7 00 00 00 A2 00 67 00 37 00 00 00 00 00 F4 .......g.7......
00001FE0   6  00 67 00 2C 00 00 00 00 00 32 00 A4 00 41 00 00 .g.,.....2...A..
00001FF0   7  00 00 01 2B 00 80 10 00 00 00 00 00 01 98 00 20 ...+........... 
00002000   8  00 00 01 24 00 00 01 B9 00 40 00 0A 00 00 00 00 ...$.....@......
00002010   9  00 EC 00 22 01 1B 01 06 00 00 00 B2 00 A0 00 00 ..."............
00002020   A  04 82 00 00 01 21 00 A3 03 00 05 10 82 00 00 40 .....!.........@
00002030   B  01 93 00 A4 00 24 01 05 02 00 69 02 01 71 00 00 .....$....i..q..
00002040   C  00 00 00 00 00 00 01 03 02 57 00 B4 0D 16 86 A6 .........W......
00002050   D  01 27 02 57 00 B4 01 0B 00 00 00 20 00 73 00 1C .'.W....... .s..
00002060   E  00 00 00 00 00 E8 00 4D 00 00 00 00 00 00 00 93 .......M........
00002070   F  00 8F 00 2A 00 00 00 00 01 2A 00 4B 00 18 00 00 ...*.....*.K....
```

```
Game          Sector 0042
Byte #         0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002080   0  00 00 00 F1 00 29 00 2C 00 00 00 00 01 00 00 AB .....).,........
00002090   1  A0 00 00 00 00 00 00 95 00 27 01 1E 08 A8 00 00 .........'......
000020A0   2  01 64 00 45 61 00 0C 83 A1 00 01 09 00 47 01 21 .d.Ea........G.!
000020B0   3  01 1C 00 00 01 C9 00 27 01 14 01 0D 00 00 01 14 .......'........
000020C0   4  00 45 03 16 10 C0 00 00 01 62 00 43 11 2C 01 0C .E.......b.C.,..
000020D0   5  00 00 00 8D 00 A1 00 1C 00 00 00 00 00 88 00 21 ...............!
000020E0   6  00 28 00 00 00 00 01 F7 00 A1 11 0E 03 19 58 00 .(............X.
000020F0   7  00 6C 00 42 00 00 04 8C 00 00 01 B6 00 21 03 28 .l.B.........!.(
00002100   8  01 01 00 00 00 A4 00 24 00 58 00 B2 00 E2 02 00 .......$.X......
00002110   9  80 02 01 00 02 58 00 C8 29 04 AA E1 01 4B 02 58 .....X..)....K.X
00002120   A  00 C8 05 1B 8C 00 01 C9 02 58 00 C8 00 00 00 00 .........X......
00002130   B  01 09 00 53 01 2A 24 8A 00 E2 00 F3 00 AF E1 30 ...S.*$........0
00002140   C  01 13 00 00 01 B8 00 4D 01 27 04 8A 00 00 00 A3 .......M.'......
00002150   D  00 A9 00 30 00 00 00 00 01 04 00 2B 01 3C 19 07 ...0.......+.<..
00002160   E  A7 C0 00 6A 00 AB 00 00 00 00 00 00 01 C5 00 26 ...j...........&
00002170   F  10 2C 00 00 00 00 01 93 00 47 00 24 00 00 00 00 .,.......G.$....
```

```
Game        Sector 0043
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002180    0 01 99 00 26 01 2C 01 0E 00 00 01 65 00 25 01 2C ...&.,.....e.%.,
00002190    1 08 A2 00 00 00 39 00 65 10 34 00 00 00 00 01 A8 .....9.e.4......
000021A0    2 00 41 01 30 03 1A 64 00 01 34 00 83 03 0E 01 27 .A.0..d..4.....'
000021B0    3 00 00 00 F8 00 43 00 18 00 00 00 00 00 D5 00 80 .....C..........
000021C0    4 00 1C 00 00 00 00 01 E7 00 83 00 0E 00 00 00 00 ................
000021D0    5 01 D5 00 81 00 00 00 00 00 00 00 58 00 B2 00 00 ...........X....
000021E0    6 00 00 00 71 02 00 4B 02 06 05 03 02 01 04 04 03 ...q..K.........
000021F0    7 03 03 02 02 03 04 06 00 20 59 4F 55 20 46 4F 55 ........ YOU FOU
00002200    8 4E 44 20 54 48 45 20 43 4F 4D 42 49 4E 41 54 49 ND THE COMBINATI
00002210    9 4F 4E 21 20 20 47 45 4E 45 52 41 4C 20 53 54 4F ON!  GENERAL STO
00002220    A 52 45 20 20 00 01 02 03 04 05 06 07 08 09 0A 0B RE  ............
00002230    B 0C 0D 0E 0F C7 F2 00 00 00 00 00 00 00 00 00 00 ................
00002240    C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002250    D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002260    E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002270    F FF 03 00 00 1E B8 01 73 00 00 00 00 00 03 00 02 .......s........
```

```
Game        Sector 0044
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002280    0 02 00 00 00 02 02 00 00 00 02 02 00 00 01 02 02 ................
00002290    1 00 00 00 02 02 00 00 00 02 02 00 00 00 02 02 00 ................
000022A0    2 00 01 02 02 00 00 00 02 02 00 00 02 02 02 00 00 ................
000022B0    3 01 33 33 00 00 01 34 34 00 00 00 35 35 00 00 00 .33...44...55...
000022C0    4 44 44 00 00 00 45 45 00 00 00 46 46 00 00 00 55 DD...EE...FF...U
000022D0    5 55 00 00 00 56 56 00 00 00 57 57 00 00 58 58 58 U...VV...WW..XXX
000022E0    6 00 00 03 02 03 02 03 02 03 02 03 02 03 02 03 02 ................
000022F0    7 02 02 02 02 02 02 01 02 03 00 00 00 01 01 01 02 ................
00002300    8 02 03 03 03 0D 0F 00 00 00 00 00 00 00 00 00 00 ................
00002310    9 14 0F 12 18 28 FB FF FE FE 00 FF FF 02 0A 32 FB ....(.........2.
00002320    A FE FE FE 00 FF FF 02 0A 32 FA FE FE FE 00 FF FF ........2.......
00002330    B 02 0F 32 FA FE FE FE 00 FE FF 02 14 3C FA FD FD ..2.........<...
00002340    C FE 00 FE FE 02 14 3C F9 FD FD FE 00 FE FE 02 14 ......<.........
00002350    D 3C F9 FD FD FE 00 FE FE 02 14 46 F8 FD FD FE 00 <.........F.....
00002360    E FE FE 02 14 46 F8 FD FD FF 00 FE FE 02 19 50 F8 ....F.........P.
00002370    F FD FD FF 00 FE FE 02 1E 44 41 47 47 45 52 20 20 ........DAGGER
```

```
Game        Sector 0045
Byte #        0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002380    0 20 20 20 20 20 20 20 04 01 FF 48 41 4E 44 20 41        ...HAND A
00002390    1 58 20 20 20 20 20 20 20 20 06 03 01 53 57 4F 52 X        ...SWOR
000023A0    2 44 20 20 20 20 20 20 20 20 20 20 08 05 01 57 49 D          ...WI
000023B0    3 5A 41 52 44 20 42 4C 41 44 45 20 20 20 08 00 FD ZARD BLADE   ...
000023C0    4 42 41 54 54 4C 45 20 41 58 20 20 20 20 20 20 0A BATTLE AX      .
000023D0    5 08 04 45 4C 56 45 4E 20 42 4C 41 44 45 20 20 20 ..ELVEN BLADE
000023E0    6 20 0C 00 05 44 57 41 52 56 45 4E 20 41 58 20 20  ...DWARVEN AX
000023F0    7 20 20 20 10 10 06 53 57 4F 52 44 20 4B 49 4E 47    ...SWORD KING
00002400    8 20 20 20 20 20 14 00 08 53 4C 49 4E 47 20 20 20      ...SLING
00002410    9 20 20 20 20 20 20 20 02 01 FF 00 00 00 53 54 4F        ......STO
00002420    A 4E 45 53 20 20 20 20 20 20 20 53 48 4F 52 54 20 NES       SHORT
00002430    B 42 4F 57 20 20 20 20 20 20 06 03 01 EC FF 01 41 BOW      ......A
00002440    C 52 52 4F 57 53 20 20 20 20 20 20 20 43 52 4F 53 RROWS       CROS
00002450    D 53 42 4F 57 20 20 20 20 20 20 20 08 06 01 EC FF SBOW       .....
00002460    E 02 51 55 41 52 52 45 4C 53 20 20 20 20 20 41 4C .QUARRELS     AL
00002470    F 45 52 54 4E 45 53 53 20 42 4F 57 20 20 06 00 04 ERTNESS BOW  ...
```

```
Game         Sector 0046
Byte #       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002480  0 00 00 00 41 52 52 4F 57 53 20 20 20 20 20 20 20 ...ARROWS
00002490  1 57 41 52 42 41 4C 4C 20 26 20 43 48 41 49 4E 14 WARBALL & CHAIN.
000024A0  2 0F 04 00 FE 00 20 20 20 20 20 20 20 20 20 20 20 .....
000024B0  3 20 20 42 4F 57 20 4F 46 20 53 54 52 45 4E 47 54   BOW OF STRENGT
000024C0  4 48 0C 00 07 EC FF 05 53 54 4F 4E 45 20 41 52 52 H......STONE ARR
000024D0  5 4F 57 53 20 20 20 20 20 20 20 20 20 20 20 20 20 OWS
000024E0  6 20 20 20 00 00 00 00 00 00 20 20 20 20 20 20 20    ......
000024F0  7 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00002500  8 20 20 20 20 20 00 00 00 00 00 00 20 20 20 20 20      ......
00002510  9 20 20 20 20 20 20 20 20 4C 45 41 54 48 45 52 20         LEATHER
00002520  A 20 20 20 20 20 20 20 02 02 FF 52 49 4E 47 20 4D        ...RING M
00002530  B 41 49 4C 20 20 20 20 20 20 04 05 01 50 4C 41 54 AIL      ...PLAT
00002540  C 45 20 4D 41 49 4C 20 20 20 20 20 06 0A 01 45 4C E MAIL     ...EL
00002550  D 56 45 4E 20 4D 41 49 4C 20 20 20 20 20 08 00 03 VEN MAIL     ...
00002560  E 57 41 52 52 49 4F 52 20 4D 41 49 4C 20 20 20 09 WARRIOR MAIL   .
00002570  F 14 05 48 45 52 4F 20 4D 41 49 4C 20 20 20 20 20 ..HERO MAIL

Game         Sector 0047
Byte #       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002580  0 20 0A 00 07 20 20 20 20 20 20 20 20 20 20 20 20  ...
00002590  1 20 20 20 00 00 00 20 20 20 20 20 20 20 20 20 20    ...
000025A0  2 20 20 20 20 20 00 00 00 53 48 49 45 4C 44 20 20      ...SHIELD
000025B0  3 20 20 20 20 20 20 20 01 01 01 4D 41 47 49 43 20        ...MAGIC
000025C0  4 41 52 4D 42 41 4E 44 53 20 02 00 FE 44 41 4E 43 ARMBANDS ...DANC
000025D0  5 49 4E 47 20 53 48 49 45 4C 44 20 03 00 05 43 4C ING SHIELD ...CL
000025E0  6 4F 41 4B 20 4F 46 20 48 49 44 49 4E 47 05 00 F8 OAK OF HIDING...
000025F0  7 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 00                .
00002600  8 00 00 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ..
00002610  9 20 00 00 00 54 4F 55 43 48 53 54 4F 4E 45 20 C8  ...TOUCHSTONE .
00002620  A C9 CA CB 01 4C 41 4E 54 45 52 4E 20 20 20 20 D2 ....LANTERN    .
00002630  B C3 7B 7B 01 50 4F 54 49 4F 4E 20 20 20 20 20 7B .{{.POTION     {
00002640  C 7B D3 B6 01 53 43 52 4F 4C 4C 20 20 20 20 20 B8 {...SCROLL     .
00002650  D 7B B9 7B FF 53 43 52 4F 4C 4C 20 20 20 20 20 B8 {.{.SCROLL     .
00002660  E 7B B9 7B FD 57 41 4E 44 20 20 20 20 20 20 20 7B {.{.WAND       {
00002670  F B4 7B B5 04 53 43 52 4F 4C 4C 20 20 20 20 20 B8 .{..SCROLL     .

Game         Sector 0048
Byte #       0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002680  0 7B B9 7B FB 53 43 52 4F 4C 4C 20 20 20 20 20 B8 {.{.SCROLL     .
00002690  1 7B B9 7B F9 4C 55 43 4B 20 53 54 4F 4E 45 20 20 {.{.LUCK STONE
000026A0  2 20 20 20 18 00 0F 48 4F 4E 49 4E 47 20 53 54 4F    ...HONING STO
000026B0  3 4E 45 20 20 20 0C 03 04 47 45 4D 20 4F 46 20 53 NE   ...GEM OF S
000026C0  4 54 52 45 4E 47 54 48 00 00 04 4F 4D 4E 49 53 43 TRENGTH...OMNISC
000026D0  5 49 45 4E 54 20 47 45 4D 20 1C 00 14 53 54 55 4D IENT GEM ...STUM
000026E0  6 42 4C 49 4E 47 20 52 4F 43 4B 20 19 00 0A 53 4F BLING ROCK ...SO
000026F0  7 4F 54 48 49 4E 47 20 4C 49 47 48 54 20 49 FA FE OTHING LIGHT I..
00002700  8 41 55 52 41 20 4F 46 20 57 41 52 44 49 4E 47 41 AURA OF WARDINGA
00002710  9 FA FF 52 41 49 4E 42 4F 57 20 53 54 52 4F 42 45 ..RAINBOW STROBE
00002720  A 20 29 02 05 50 41 54 48 20 4C 49 47 48 54 45 4E  )..PATH LIGHTEN
00002730  B 45 52 20 44 FA FC 53 4D 55 44 47 45 20 50 4F 54 ER D..SMUDGE POT
00002740  C 20 20 20 20 20 40 FA 0A 46 4F 52 54 49 46 59 49      @..FORTIFYI
00002750  D 4E 47 20 42 52 45 57 14 01 02 52 45 46 4C 45 58 NG BREW...REFLEX
00002760  E 20 44 52 41 55 47 48 54 20 10 01 02 4C 49 47 48  DRAUGHT ...LIGH
00002770  F 54 46 4F 4F 54 20 4C 41 47 45 52 18 01 04 4D 45 TFOOT LAGER...ME
```

```
Game          Sector 0049
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002780   0 44 49 43 49 4E 41 4C 20 41 4C 45 20 20 05 03 14 DICINAL ALE  ...
00002790   1 41 43 49 44 20 20 20 20 20 20 20 20 20 20 20 01 ACID           .
000027A0   2 01 05 4D 4F 52 41 4C 45 20 20 20 20 20 20 20 20 ..MORALE
000027B0   3 20 4D 04 02 50 52 4F 54 45 43 54 49 4F 4E 20 20  M..PROTECTION
000027C0   4 20 20 20 51 04 02 45 53 50 20 20 20 20 20 20 20    Q..ESP
000027D0   5 20 20 20 20 20 74 04 00 52 45 50 41 49 52 20 20      t..REPAIR
000027E0   6 20 20 20 20 20 20 20 08 03 FE 4D 45 4D 4F 52 59        ...MEMORY
000027F0   7 20 41 42 53 4F 52 42 45 52 1D 01 0A 46 49 52 45  ABSORBER...FIRE
00002800   8 42 41 4C 4C 20 20 20 20 20 20 20 6D 06 08 53 4E BALL       m..SN
00002810   9 41 52 45 20 20 20 20 20 20 20 20 20 20 61 08 03 ARE          a..
00002820   A 46 49 4E 44 20 54 52 41 50 53 20 20 20 20 20 78 FIND TRAPS     x
00002830   B 04 00 48 45 41 4C 54 48 20 53 48 45 41 54 48 20 ..HEALTH SHEATH
00002840   C 20 05 03 1E 4D 41 44 20 53 43 52 49 42 42 4C 45  ...MAD SCRIBBLE
00002850   D 20 20 20 21 01 02 4C 49 47 48 54 4E 49 4E 47 20    !..LIGHTNING
00002860   E 52 4F 44 20 20 71 06 0F 57 45 41 50 4F 4E 20 48 ROD  q..WEAPON H
00002870   F 4F 41 52 44 45 52 20 34 08 00 43 4F 4E 53 55 4D OARDER 4..CONSUM

Game          Sector 004A
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002880   0 49 4E 47 20 42 45 41 4D 20 71 04 9C 53 54 41 46 ING BEAM q..STAF
00002890   1 46 20 4F 46 20 4C 49 46 45 20 20 29 02 14 42 41 F OF LIFE  )..BA
000028A0   2 43 4B 46 49 52 49 4E 47 20 42 45 41 4D 04 01 14 CKFIRING BEAM...
000028B0   3 4C 49 47 48 54 4E 49 4E 47 20 20 20 20 20 20 71 LIGHTNING      q
000028C0   4 06 1E 48 41 53 54 45 20 20 20 20 20 20 20 20 20 ..HASTE
000028D0   5 20 3C 04 01 56 55 4C 4E 45 52 41 42 49 4C 49 54  <..VULNERABILIT
000028E0   6 59 20 20 65 03 05 47 4F 4F 44 57 49 4C 4C 20 20 Y  e..GOODWILL
000028F0   7 20 20 20 20 20 5C 03 04 44 45 53 50 41 49 52 20      \..DESPAIR
00002900   8 20 20 20 20 20 20 20 19 01 14 44 49 53 49 4E 54        ...DISINT
00002910   9 45 47 52 41 54 45 20 20 20 6D 02 64 4E 55 4C 4C EGRATE   m.dNULL
00002920   A 49 46 59 20 50 4F 57 45 52 20 20 59 04 32 53 55 IFY POWER  Y.2SU
00002930   B 50 50 52 45 53 53 20 54 52 41 50 53 20 79 06 00 PPRESS TRAPS y..
00002940   C 52 45 53 54 4F 52 41 54 49 4F 4E 20 20 20 20 05 RESTORATION    .
00002950   D 04 64 4D 45 4D 4F 52 59 20 44 52 41 49 4E 20 20 .dMEMORY DRAIN
00002960   E 20 21 01 04 05 18 00 1C 04 19 01 1D 00 00 28 2C  !............(,
00002970   F 2E 32 5A 5E 60 64 00 00 2D 33 36 3C 5A 5E 60 64 .2Z^`d..-36<Z^`d

Game          Sector 004B
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002980   0 00 00 32 3A 3E 46 5A 5F 5F 64 00 00 3C 44 48 50 ..2:>FZ__d..<DHP
00002990   1 5A 5F 5F 64 00 00 4B 49 4E 47 20 20 20 20 20 20 Z__d..KING
000029A0   2 20 02 0A CC C5 CD C6 00 0C 52 41 49 4E 42 4F 57  ........RAINBOW
000029B0   3 20 4F 52 42 00 0A CF C7 BF B7 00 0E 20 20 20 20  ORB........
000029C0   4 20 20 20 20 20 20 20 00 00 00 00 00 00 00 00 20        ........
000029D0   5 20 20 20 20 20 20 20 20 20 20 00 00 00 00 00 00           ......
000029E0   6 00 00 20 20 20 20 20 20 20 20 20 20 20 00 00 00 ..           ...
000029F0   7 00 00 00 00 00 20 20 20 20 20 20 20 20 20 20 20 .....
00002A00   8 00 00 00 00 00 00 00 00 20 20 20 20 20 20 20 20 ........
00002A10   9 20 20 20 00 00 00 00 00 00 00 00 20 20 20 20 20    ........
00002A20   A 20 20 20 20 20 20 00 00 00 00 00 00 00 00 A8 AC       ..........
00002A30   B A8 B0 B4 04 B0 20 A0 9C 98 A4 B8 01 B4 20 4F 50 ..... ....... OP
00002A40   C 45 4E 20 20 20 20 45 58 50 4C 4F 53 49 4F 4E 20 EN    EXPLOSION
00002A50   D 28 0A 44 45 41 44 46 41 4C 4C 20 20 04 1E 50 49 (.DEADFALL  ..PI
00002A60   E 54 20 54 52 41 50 20 20 04 32 41 43 49 44 20 53 T TRAP  .2ACID S
00002A70   F 50 52 41 59 09 37 46 55 4E 47 55 53 20 20 20 20 PRAY.7FUNGUS
```

```
Game            Sector 004C
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002A80   0 01 41 46 4F 52 47 45 54 20 47 41 53 1D 4B 4D 45 .AFORGET GAS.KME
00002A90   1 54 41 4C 20 4D 4F 4C 44 0D 50 43 55 52 53 45 20 TAL MOLD.PCURSE 
00002AA0   2 20 20 20 20 15 55 54 52 41 4E 53 4D 55 54 45 20     .UTRANSMUTE 
00002AB0   3 31 5F 43 55 52 53 45 20 20 20 20 20 11 64 BF 7B 1_CURSE     .d.{
00002AC0   4 96 6B 6F 79 EF 1E 6F 17 A9 7A 26 6A 6F 72 F9 1F .koy..o..z&jor..
00002AD0   5 69 17 25 A2 C6 62 6F AC F5 1F 65 1A 52 A5 46 65 i.%..bo...e.R.Fe
00002AE0   6 6F A4 F2 1F 62 1B 8C D8 6E E8 E1 D6 FC 1F EC 15 o...b...n.......
00002AF0   7 4E DE CE DE 13 13 13 13 13 13 13 13 1E 1E EE EE N...............
00002B00   8 1E 1E 1E 1E 1E 1A 17 12 1A 16 1E 1E 16 00 1E 6E ...............n
00002B10   9 00 3C 66 C3 81 15 09 09 1F 07 03 01 0F 13 2D 53 .<f...........-S
00002B20   A 00 3C 66 C3 81 A8 90 90 F8 E0 C0 80 F0 C8 B4 CA .<f.............
00002B30   B 00 01 00 01 02 04 01 01 01 02 07 05 05 3F 2A 3F .............?*?
00002B40   C 80 88 90 E0 80 80 40 40 40 20 F0 50 50 FE AA FE ......@@@ .PP...
00002B50   D 00 00 00 00 00 03 01 00 1C 0E 07 E3 70 39 1D 01 ............p9..
00002B60   E 00 00 70 38 1C 8E C0 E7 77 07 3F BF 3F FF FF FF ..p8....w.?.?...
00002B70   F 0F 0F 01 1D 39 70 E3 07 0E 1C 00 01 03 00 00 00 ....9p..........
```

```
Game            Sector 004D
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002B80   0 FF FF FF FF FF 3F BF 3F 07 77 E7 C0 8E 1C 38 70 .....?.?.w....8p
00002B90   1 00 60 F0 F1 63 05 31 79 7A 32 78 FF AA AA AA FF .`..c.1yz2x.....
00002BA0   2 20 7C 20 84 CE A4 84 84 44 40 00 FE AA AA AA FE  | .....D@......
00002BB0   3 00 0F 18 30 60 60 63 27 27 63 60 60 30 18 0F 00 ...0``c''c``0...
00002BC0   4 00 E0 30 18 0C 04 04 94 94 14 04 0C 18 30 E0 00 ..0..........0..
00002BD0   5 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002BE0   6 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002BF0   7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002C00   8 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002C10   9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002C20   A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002C30   B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002C40   C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002C50   D 00 00 03 0F 1F 37 23 23 03 07 0F 0E 0C 0C 0C 00 .....7##........
00002C60   E 00 00 C0 F0 F8 EC C4 C4 C0 E0 F0 70 30 30 30 00 ...........p000.
00002C70   F 00 00 11 1E 10 13 11 08 08 08 04 04 02 01 00 00 ................
```

```
Game            Sector 004E
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002C80   0 00 00 88 78 08 C8 88 10 10 10 20 20 40 80 00 00 ...x......  @...
00002C90   1 FF 80 80 80 BF A8 A8 A8 FE 88 8F 80 80 FE AA FF ................
00002CA0   2 FF 01 71 41 C1 41 79 21 21 39 E9 49 49 79 01 FF ..qA.Ay!!9.IIy..
00002CB0   3 00 00 00 00 3F 00 00 00 00 08 22 14 C1 14 22 08 ....?....."...".
00002CC0   4 81 99 81 24 18 24 81 7E 49 B1 43 43 87 0D 39 FF ...$.$.~I.CC..9.
00002CD0   5 FF CF C0 CA C5 C0 CF FF FF F3 03 A3 53 03 F3 FF ............S...
00002CE0   6 FF 03 30 4F 20 4F 30 00 FF C0 0C F2 04 F2 0C 00 ..0O O0.........
00002CF0   7 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002D00   8 00 00 00 00 00 00 00 00 FF 39 8D 07 63 93 89 89 .........9..c...
00002D10   9 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002D20   A 00 00 00 00 00 00 00 00 99 C3 A5 81 81 A5 C3 FF ................
00002D30   B FF C3 E7 C3 81 81 C3 FF E0 C6 D0 E1 E4 F3 F8 FF ................
00002D40   C 07 63 0B 87 27 CF 1F FF 92 8D C2 C2 E1 B0 9C FF .c..'...........
00002D50   D FF F4 E4 C6 8F FA 9C 88 88 9C FA 8F C6 E4 F4 FF ................
00002D60   E FF 2F 27 63 F1 5F 39 11 11 39 5F F1 63 27 2F FF ./'c._9..9_.c'/.
00002D70   F FF FC E0 CB D9 E6 EC E0 FF 3F 07 D3 9B 67 37 07 .........?...g7.
```

```
Game          Sector 004F
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002D80   0 00 00 00 00 00 00 00 00 FF 9C B1 E0 C6 C9 91 91 ................
00002D90   1 00 00 0F 1E 3C 73 67 FE 00 00 F0 78 3C CE E6 7F ....<sg....x<...
00002DA0   2 00 00 18 24 24 18 7E FF 00 00 00 00 00 3C 18 7E ...$$.~......<.~
00002DB0   3 00 00 00 07 0E 1D 7F 00 00 00 30 90 F8 FC FE 00 ..........0.....
00002DC0   4 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002DD0   5 3C 3C 3C 3C 3C 3C 3C 3C 00 00 FF FF FF FF 00 00 <<<<<<<<........
00002DE0   6 24 18 18 24 24 18 18 24 00 00 99 66 66 99 00 00 $..$$..$...ff...
00002DF0   7 FF FF C3 C3 C3 C3 FF FF AA 55 AA 55 AA 55 AA 55 .........U.U.U.U
00002E00   8 3C 3C FF FF FF FF 00 00 3C 3C 3F 3F 3F 3F 3C 3C <<......<<????<<
00002E10   9 10 18 18 18 18 18 18 18 18 7E DB 99 18 24 18 00 .........~...$..
00002E20   A 18 3C 0E 26 46 86 0C FF 0C 86 46 26 0E 3C 18 00 .<.&F.....F&.<..
00002E30   B 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002E40   C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002E50   D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002E60   E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002E70   F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................

Game          Sector 0050
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002E80   0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00002E90   1 80 81 82 83 84 85 86 87 E0 E1 7B 7B E2 E3 7B 7B ..........{{..{{
00002EA0   2 AC AD AE AF A8 A9 AA AB D0 BA D1 BB C4 7B 7B 7B .............{{{
00002EB0   3 B0 B1 B2 B3 7B D4 7B D5 01 03 00 01 A9 42 43 46 ....{.{......BCF
00002EC0   4 57 4C 4D 4F 4B 54 55 4E 00 00 00 00 00 02 14 1E WLMOKTUN........
00002ED0   5 46 4F 55 4E 54 41 49 4E 20 20 20 20 20 20 20 20 FOUNTAIN
00002EE0   6 4C 49 56 49 4E 47 20 53 54 41 54 55 45 20 20 20 LIVING STATUE
00002EF0   7 47 4F 4C 44 20 50 49 45 43 45 53 20 20 20 20 20 GOLD PIECES
00002F00   8 20 20 4D 41 47 49 43 41 4C 20 49 54 45 4D 53 20   MAGICAL ITEMS
00002F10   9 50 4C 41 59 45 52 20 20 20 20 20 20 50 41 52 54 PLAYER      PART
00002F20   A 59 20 20 20 20 20 20 20 45 58 50 45 52 49 45 4E Y       EXPERIEN
00002F30   B 43 45 20 20 4C 45 56 45 4C 20 20 20 20 20 20 20 CE  LEVEL
00002F40   C 52 41 4E 47 45 44 20 20 20 20 20 20 57 45 41 50 RANGED      WEAP
00002F50   D 4F 4E 20 20 20 20 20 20 41 52 4D 4F 52 20 20 20 ON      ARMOR
00002F60   E 20 20 20 20 53 48 49 45 4C 44 20 20 20 20 20 20     SHIELD
00002F70   F 4D 4F 4E 53 54 45 52 20 20 20 20 20 54 52 41 50 MONSTER     TRAP

Game          Sector 0051
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00002F80   0 20 20 20 20 20 20 20 20 52 41 54 49 4F 4E 20 20         RATION
00002F90   1 20 20 20 20 48 45 41 4C 49 4E 47 20 20 20 20 20     HEALING
00002FA0   2 53 50 45 45 44 20 20 20 20 20 20 20 52 45 53 49 SPEED       RESI
00002FB0   3 53 54 41 4E 43 45 20 20 4D 41 47 49 43 20 20 20 STANCE  MAGIC
00002FC0   4 20 20 20 20 41 4C 4C 20 20 20 20 20 20 20 20 20     ALL
00002FD0   5 50 52 4F 54 45 43 54 49 4F 4E 20 20 4D 4F 42 49 PROTECTION  MOBI
00002FE0   6 4C 49 54 59 20 20 20 20 53 50 45 43 49 41 4C 20 LITY    SPECIAL
00002FF0   7 20 20 20 20 43 4C 41 53 53 20 20 20 20 20 20 20     CLASS
00003000   8 41 54 54 41 43 4B 20 20 20 20 20 20 44 41 4D 41 ATTACK      DAMA
00003010   9 47 45 20 20 20 20 20 20 43 4F 4D 42 41 54 20 20 GE      COMBAT
00003020   A 20 20 20 20 52 4F 4F 4D 20 20 20 20 20 20 20 20     ROOM
00003030   B 41 56 41 49 4C 41 42 49 4C 49 54 59 49 4E 46 4F AVAILABILITYINFO
00003040   C 52 4D 41 54 49 4F 4E 20 50 4F 57 45 52 20 43 48 RMATION POWER CH
00003050   D 41 4E 43 45 50 52 4F 42 41 42 49 4C 49 54 59 20 ANCEPROBABILITY
00003060   E 42 52 49 42 41 42 49 4C 49 54 59 20 47 4F 4C 44 BRIBABILITY GOLD
00003070   F 20 20 20 20 20 20 20 20 48 49 54 20 50 4F 49 4E         HIT POIN
```

```
Game          Sector 0052
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00003080   0 54 53 20 20 57 41 4E 44 45 52 49 4E 47 20 20 20 TS  WANDERING
00003090   1 4C 55 43 4B 20 20 20 20 20 20 20 20 43 4F 4E 53 LUCK        CONS
000030A0   2 55 4D 50 54 49 4F 4E 20 42 4F 4E 55 53 20 20 20 UMPTION BONUS
000030B0   3 20 20 20 20 49 4E 54 45 52 56 41 4C 20 20 20 20     INTERVAL
000030C0   4 30 55 20 30 4C 20 30 36 47 30 35 4C 30 36 59 30 0U 0L 06G05L06Y0
000030D0   5 35 59 30 57 20 30 32 20 30 33 20 2B 2B 2B 46 30 5Y0W 02 03 +++F0
000030E0   6 4C 31 54 20 31 41 20 31 35 4F 2B 2B 2B 31 4D 43 L1T 1A 15O+++1MC
000030F0   7 56 38 52 41 58 5A 31 42 5A 38 36 47 38 4B 4A 38 V8RAXZ1BZ86G8KJ8
00003100   8 4B 4C 38 49 51 38 53 20 38 48 20 38 45 44 38 4D KL8IQ8S 8H 8ED8M
00003110   9 43 46 38 55 38 55 20 4E 38 50 2B 2B 2B 2B 2B 2B CF8U8U N8P++++++
00003120   A 43 48 45 53 54 20 20 20 20 20 20 20 56 41 55 4C CHEST       VAUL
00003130   B 54 20 20 20 20 20 20 20 6B 6B 6B 6B 6B 6B 6B 6B T       kkkkkkkk
00003140   C 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B kkkkkkkkkkkkkkkk
00003150   D 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 6B 6B 6B 6B 6B kkk    kkkkkkkkk
00003160   E 6B 67 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B kgkkkkkkkkkkkkkk
00003170   F 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 6B 6A 60 60 60 kkk    kkkkkj```
```

```
Game          Sector 0053
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00003180   0 67 63 60 60 60 67 60 60 60 60 67 6B 6B 6B 6B 6B gc```g````gkkkkk
00003190   1 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 6B 61 6B 6B 67 kkk    kkkkkakkg
000031A0   2 60 60 60 60 60 65 6B 6B 6B 6B 66 67 60 60 60 67 `````ekkkkfg```g
000031B0   3 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 67 63 60 60 60 kkk    kkkkgc```
000031C0   4 60 60 60 60 60 65 6B 6B 6B 6B 61 6B 6B 6B 6B 61 `````ekkkkakkkka
000031D0   5 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 61 6B 6B 6B 6B kkk    kkkkakkkk
000031E0   6 6B 6B 6B 6B 6B 61 6B 6B 6B 6B 61 6B 6B 6B 6B 61 kkkkkakkkkakkkka
000031F0   7 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 61 6B 6B 6B 6B kkk    kkkkakkkk
00003200   8 6B 6B 6B 6B 6B 61 6B 6B 6B 6B 61 6B 6B 6B 6B 61 kkkkkakkkkakkkka
00003210   9 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 61 6B 6B 6B 6B kkk    kkkkakkkk
00003220   A 6B 6B 6B 6B 6B 61 6B 6B 6B 6B 61 6B 6B 6B 6B 61 kkkkkakkkkakkkka
00003230   B 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 66 60 60 60 60 kkk    kkkkf````
00003240   C 60 60 60 67 60 65 6B 6B 6B 6B 67 60 67 6B 6B 61 ```g`ekkkkg`gkka
00003250   D 6B 6B 6B 20 20 20 20 6B 67 60 60 65 6B 6B 6B 6B kkk    kg``ekkkk
00003260   E 6B 6B 6B 6B 6B 67 60 60 60 60 60 60 65 6B 6B 6A kkkkkg``````ekkj
00003270   F 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 68 6B 6B 6B 6B kkk    kkkkhkkkk
```

```
Game          Sector 0054
Byte #          0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00003280   0 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 61 6B 6B 61 kkkkkkkkkkkkakka
00003290   1 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 66 60 60 60 60 kkk    kkkkf````
000032A0   2 60 60 60 60 60 60 60 60 67 60 60 60 65 6B 6B 67 ````````g```ekkg
000032B0   3 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 66 60 60 60 60 kkk    kkkkf````
000032C0   4 60 60 60 60 67 60 60 60 60 67 6B 6B 61 6B 6B 6B ````g````gkkakkk
000032D0   5 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 68 60 60 60 60 kkk    kkkkh````
000032E0   6 60 60 60 60 60 60 60 60 60 60 60 60 65 6B 6B 6B ````````````ekkk
000032F0   7 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 6B 6B 67 60 60 kkk    kkkkkkg``
00003300   8 60 60 67 6B 6B 6B 6B 6B 6B 6B 6B 6B 61 6B 6B 6B ``gkkkkkkkkkakkk
00003310   9 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 6B 6B 6B 6B 6B kkk    kkkkkkkkk
00003320   A 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 67 6B 6B 6B kkkkkkkkkkkkgkkk
00003330   B 6B 6B 6B 20 20 20 20 6B 6B 6B 6B 6B 6B 6B 6B 6B kkk    kkkkkkkkk
00003340   C 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B 6B kkkkkkkkkkkkkkkk
00003350   D 6B 6B 06 0A 04 00 80 80 33 00 00 03 43 53 31 20 kk......3...CS1
00003360   E 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00003370   F 20 20 20 20 20 20 20 20 0B 09 0B 0B 0D 09 0D 0B         ........
```

Graphics Bank #3(?) Compass Directions, Map Location & Weapon Cursor Graphics (fixed locations)

Game Byte #	Sector 0022 Bytes		Char-Codes
00000080	**00-1F**	Graphic for 'N'orth	80-83
000000A0	**20-3F**	Graphic for 'E'ast	84-87
000000C0	**40-5F**	Graphic for 'S'outh	88-8B
000000E0	**60-7F**	Graphic for 'W'est	8D-8F
00000100	**80-9F**	Party location indicator on map	90-93
00000120	**A0-BF**	Ranged Weapon/ Magical cursor graphic	94-97

Ranged Weapon & Magical Attack Graphic Sequences (Codes 98-BF) (as utilized in Quest)

00000140	**C0-DF**	Flying blade graphic #1 (***invoked by spells #70 & 71***)	98-9B
00000160	**E0-FF**	Flying blade graphic #2	9C-9F

Game Byte #	Sector 0023 Bytes		
00000180	**00-1F**	Flying blade graphic #3	A0-A3
000001A0	**20-3F**	Flying blade graphic #4	A4-A7
000001C0	**40-5F**	Ranged weapon projectile graphic	A8-AB
000001E0	**60-7F**	Ranged Weapon Projectile	AC-AF
00000200	**80-9F**	Ranged Weapon Projectile	B0-B3
00000220	**A0-BF**	Monster Magical Attack	B4-B7
00000240	**C0-DF**	Impact from Ranged Attack	B8-BB
00000260	**E0-FF**	Bank space (Quest)	BC-BF

Note: The memory used to define Char-Codes 98 - BF is completely re-definable for both Ranged and Magical Weapons sequences. See Sector 004B, Bytes AE - BD for allocating details.

Game Title and Description (First 8 lines of text, 32 columns each)

Game Byte #	Sector 0024 Bytes	
00000280	**00-1F**	1st line of text/ graphics (i.e. Game Title)
000002A0	**20-3F**	2nd line of text
000002C0	**40-5F**	3rd line of text
000002E0	**60-7F**	4th line of text
00000300	**80-9F**	5th line of text
00000320	**A0-BF**	6th line of text
00000340	**C0-DF**	7th line of text
00000360	**E0-FF**	8th line of text

Game Title and Description (Last 4 lines of text, 32 columns each)

Game Byte #	Sector 0025 Bytes		
00000380	**00-1F**	9th line of text/ graphics	
000003A0	**20-3F**	10th line of text	
000003C0	**40-5F**	11th line of text	
000003E0	**60-7F**	12th line of text	
00000400	**80-9F**	Not used \	
00000420	**A0-BF**	Not used \	You may place Game developer information, date of
00000440	**C0-DF**	Not used /	completion, game version #, etc. here as a form of
00000460	**E0-FF**	Not used /	REM Statements.

Note: The first 8 lines of the Game Description Screen are reserved by the module for the ***'Tunnels of Doom'*** title and status reports on constructing/ stocking of new dungeons. The module accepts 12 lines of text, 32 columns each of any character in the 00 - 5F range. The first line of text is typically used for the Game Title and this is what is displayed at the end of a game if you have successfully completed the same, along with any Quest objects you successfully recovered.

Char-Codes 00-1F are the Defense and Attack graphics of up to 4 Characters, listed in the order that they were selected for play from the last saved game. (Also available if "Continue Current Game" is selected.) 20-5F are the same as those from Sectors 0027 & 0028. Above Char-Code 5F colored square blocks are primarily available:

Orange from 60-6F. Green 70-77. Blue 78-7F. Magenta 80-87. Red 88-E7. Typically A0 is used to provide the colored border. Codes F0-F3 will display the Defense graphic and F8-FB the Attack graphic of the last Monster encountered in a saved game. F4-F7 the stairs down of saved game. E8-EF Blue messy graphics. If no monsters were encountered before the game was save, F0-F7 displays yellow squares. Bytes 80-FF of Sector 0025 are unused, but no additional text is accepted. It can be used as a **REM** area for Game Developer information, date, game version etc.

Saved Character Graphics – Common Graphics Bank (common to all three >7F Banks)

Game	Sector 0026 (Defines up to 4 Players from Current or last Saved Game)		
Byte #	Bytes		Char-Codes
00000480	00-1F	Character #1 Defense pose	00-03
000004A0	20-3F	Character #1 Attack pose	04-07
000004C0	40-5F	Character #2 Defense pose	08-0B
000004E0	60-7F	Character #2 Attack pose	0C-0F
00000500	80-9F	Character #3 Defense pose	10-13
00000520	A0-BF	Character #3 Attack pose	14-17
00000540	C0-DF	Character #4 Defense pose	18-1B
00000560	E0-FF	Character #4 Attack pose	1C-1F

Note: These Character graphics are used if you select, "Continue Current Game" or, they are offered for use if the number of Players selected is the same as those of the last saved game.

Redefined ASCII Characters 32-63 (As Char-Codes)

Game	Sector 0027		Colors set by Byte: 74	Game	Sector 0027		Colors set by Byte: 76
Byte #	Bytes		Char-Codes	Byte #	Bytes		Char-Codes
00000580	00-07	(space)	20	00000600	80-87	0	30
00000588	08-0F	!	21	00000608	88-8F	1	31
00000590	10-17	"	22	00000610	90-97	2	32
00000598	18-1F	#	23	00000618	98-9F	3	33
000005A0	20-27	$	24	00000620	A0-A7	4	34
000005A8	28-2F	%	25	00000688	A8-AF	5	35
000005B0	30-37	&	26	00000630	B0-B7	6	36
000005B8	38-3F	'	27	00000638	B8-BF	7	37
			Colors set by Byte: 75				Colors set by Byte: 77
000005C0	40-47	(	28	00000640	C0-C7	8	38
000005C8	48-4F	)	29	00000648	C8-CF	9	39
000005D0	50-57	*	2A	00000650	D0-D7	:	3A
000005D8	58-5F	+	2B	00000658	D8-DF	;	3B
000005E0	60-67	,	2C	00000660	E0-E7	↑ (up arrow)	3C
000005E8	68-6F	-	2D	00000668	E8-EF	=	3D
000005F0	70-77	.	2E	00000670	F0-F7	↓ (down arrow)	3E
000005F8	78-7F	/	2F	00000678	F8-FF	?	3F

Redefined ASCII Characters 64-95 (As Char-Codes)

Game	Sector 0028		Colors set by Byte: 78	Game	Sector 0028		Colors set by Byte: 7A
Byte #	Bytes		Char-Codes	Byte #	Bytes		Char-Codes
00000680	00-07	@	40	00000700	80-87	P	50
00000688	08-0F	A	41	00000708	88-8F	Q	51
00000690	10-17	B	42	00000710	90-97	R	52
00000698	18-1F	C	43	00000718	98-9F	S	53
000006A0	20-27	D	44	00000720	A0-A7	T	54
000006A8	28-2F	E	45	00000728	A8-AF	U	55
000006B0	30-37	F	46	00000730	B0-B7	V	56
000006B8	38-3F	G	47	00000738	B8-BF	W	57
			Colors set by Byte: 79				Colors set by Byte: 7B
000006C0	40-47	H	48	00000740	C0-C7	X	58
000006C8	48-4F	I	49	00000748	C8-CF	Y	59
000006D0	50-57	J	4A	00000750	D0-D7	Z	5A
000006D8	58-5F	K	4B	00000758	D8-DF	©	5B
000006E0	60-67	L	4C	00000760	E0-E7	→ (right arrow)	5C
000006E8	68-6F	M	4D	00000768	E8-EF	_/ (cursor space)	5D
000006F0	70-77	N	4E	00000770	F0-F7	\■/ (cursor)	5E
000006F8	78-7F	O	4F	00000778	F8-FF	━ (thick bar)	5F

Note: The Char-Codes 20 - 5F are provided by the module. On game boot-up they are transferred from the module to VDP memory and then saved to these Sectors with a 'Saved Game'. If redefined they will revert back to these sets.

Map Symbol Graphics (ASCII Characters 96-127)

Map Graphics/ Colors - Explored Areas

Game	Sector 0029	Colors set by Byte: 7C	
Byte #	Bytes		Char-Codes
00000780	00-07	═	60
00000788	08-0F	‖	61
00000790	10-17	╬	62
00000798	18-1F	╩	63
000007A0	20-27	╦	64
000007A8	28-2F	╣	65
000007B0	30-37	╠	66
000007B8	38-3F	▣ Room Symbol	67
		Colors set by Byte: 7E	
000007C0	40-47	⇧ Stairs Up	68
000007C8	48-4F	⇩ Stairs Down	69
000007D0	50-57	⊕ Hall Fountain	6A
000007D8	58-5F	Floor & Map Texture	6B
000007E0	60-67	→ Right arrow	6C
000007E8	68-6F	_/ Cursor Space	6D
000007F0	70-77	Blank	6E
000007F8	78-7F	Blank	6F

Map Graphics/ Colors - Unexplored (Default = Gray on Gray = Invisible)

Game	Sector 0029	Colors set by Byte: 7D	
Byte #	Bytes		Char-Codes
00000800	80-87	═	70
00000808	88-8F	‖	71
00000810	90-97	╬	72
00000818	98-9F	╩	73
00000820	A0-A7	╦	74
00000828	A8-AF	╣	75
00000830	B0-B7	╠	76
00000838	B8-BF	▣ Room Symbol	77
		Colors set by Byte: 7F	
00000840	C0-C7	⇧ Stairs Up	78
00000848	C8-CF	⇩ Stairs Down	79
00000850	D0-D7	⊕ Hall Fountain	7A
00000858	D8-DF	Floor & Map Texture	7B
00000860	E0-E7	→ Right arrow	7C
00000868	E8-EF	_/ Cursor Space	7D
00000870	F0-F7	Blank	7E
00000878	F8-FF	Blank	7F

Note: Map Graphic sets are duplicated so that explored & unexplored (generally not visible) areas can be different. When a map is found the colors turn blue on gray.

Hallway Graphics – Graphics Bank #1 (>7F Character-Codes)

Game	Sector 002A		
Byte #	Bytes	Colors set in Sector 4C, Bytes: 3E, 48, 52, 5C & 66 (Each byte sets color for 2 Floors)	Char-Codes
00000880	00-07	Walls to your R & L, but not directly in front/ above doors	80
00000888	08-0F	Floor of hallways, but not floor/ wall interface	81
00000890	10-17	L floor/ wall interface	82
00000898	18-1F	R floor/ wall interface	83
000008A0	20-27	Blank	84
000008A8	28-2F	Blank	85
000008B0	30-37	Blank	86
000008B8	38-3F	Blank	87
		Colors set in Sector 4C, Bytes: 3F, 49, 53, 5D & 67 (Each byte sets color for 2 Floors)	
000008C0	40-47	Ceiling, but not along wall edges	88
000008C8	48-4F	L Ceiling/ wall interface, proximal - more wall than ceiling	89
000008D0	50-57	L Ceiling/ wall interface, distal - more ceiling than wall	8A
000008D8	58-5F	R Ceiling/ wall interface, proximal - more wall than ceiling	8B
000008E0	60-67	R Ceiling/ wall interface, distal - more ceiling than wall	8C
000008E8	68-6F	Blank	8D
000008F0	70-77	Blank	8E
000008F8	78-7F	Blank	8F
		Colors set in Sector 4C, Bytes: 40, 4A, 54, 5E & 68 (Each byte sets color for 2 Floors)	
00000900	80-87	Wall viewed directly in front (not to either side)	90
00000908	88-8F	Wall and door viewed 4 paces away	91
00000910	90-97	Wall and door viewed 4 paces away	92
00000918	98-9F	Blank	93
00000920	A0-A7	Blank	94
00000928	A8-AF	Blank	95
00000930	B0-B7	Blank	96
00000938	B8-BF	Blank	97
		Colors set in Sector 4C, Bytes: 41, 4B, 55, 5F & 69 (Each byte sets color for 2 Floors)	
00000940	C0-C7	L corner, above Door \	98
00000948	C8-CF	1st of Center blocks, above Door \	99
00000950	D0-D7	2nd of Center blocks, above Door / when door is on your L	9A
00000958	D8-DF	R corner, above Door /	9B
00000960	E0-E7	R corner, above Door \	9C
00000968	E8-EF	1st of Center blocks, above Door \	9D
00000970	F0-F7	2nd of Center blocks, above Door / when door is on your R	9E
00000978	F8-FF	L corner, above Door /	9F

Hallway Graphics (Especially Door and Hall Fountains)

Game	Sector 002B		
Byte #	Bytes	Colors set in Sector 4C, Bytes: 42, 4C, 56, 60 & 6A (Each byte sets color for 2 Floors)	Char-Codes
00000980	00-07	All of Door 3-paces away	A0
00000988	08-0F	Blank	A1
00000990	10-17	Blank	A2
00000998	18-1F	Doorknob, 2 & 1-paces away	A3
000009A0	20-27	Top R corner of Door, when Door is on your L	A4
000009A8	28-2F	Top L corner of Door, when Door is on your R	A5
000009B0	30-37	Top of Door, viewed directly ahead, 1-pace away	A6
000009B8	38-3F	Bottom of Door, viewed directly ahead, 1-pace away	A7
		Colors set in Sector 4C, Bytes: 43, 4D, 57, 61 & 6B (Each byte sets color for 2 Floors)	
000009C0	40-47	Border of Walls, directly ahead, viewed from 2-paces	A8
000009C8	48-4F	Blank	A9
000009D0	50-57	Blank	AA
000009D8	58-5F	Blank	AB
000009E0	60-67	Blank	AC
000009E8	68-6F	Blank	AD
000009F0	70-77	Blank	AE
000009F8	78-7F	Blank	AF
		Colors set in Sector 4C, Bytes: 44, 4E, 58, 62 & 6C (Each byte sets color for 2 Floors)	
00000A00	80-87	Center of Top of Fountain, viewed 3, 2 & 1-paces away	B0
00000A08	88-8F	Center row of Fountain Top, Design,(4 of them), 1-pace away	B1
00000A10	90-97	L half of Fountain 4-paces away, L Top of Fountain 3-paces, L Top corner 2 & 1-paces	B2
00000A18	98-9F	R half of Fountain 4-paces away, R Top of Fountain 3-paces, R Top corner 2 & 1-paces	B3
00000A20	A0-A7	Blank	B4
00000A28	A8-AF	Blank	B5
00000A30	B0-B7	Blank	B6
00000A38	B8-BF	Blank	B7
		Colors set in Sector 4C, Bytes: 45, 4F, 59, 63 & 6D (Each byte sets color for 2 Floors)	
00000A40	C0-C7	Base of Fountain - 3, 2 & 1 paces away (mixed with floor background 2 & 1-paces away)	B8
00000A48	C8-CF	L bottom corner, of Top of Fountain, 2 & 1-paces away	B9
00000A50	D0-D7	R bottom corner, of Top of Fountain, 2 & 1-paces away	BA
00000A58	D8-DF	Top row of Center of Fountain, 1-pace away (Quest = solid bar)	BB
00000A60	E0-E7	Blank	BC
00000A68	E8-EF	Blank	BD
00000A70	F0-F7	Blank	BE
00000A78	F8-FF	Blank	BF

Note: There are two sets of Graphic Codes in the 80 - FF range. The first is used for Hallways, outside of Door and Fountain Graphics. The second set defines Room Graphics and their contents: chests, vaults, all found items, monsters present, as well as graphics for the room, doors and room corners. When a Monster is encountered in the Hallway, the game changes to this Char-Code Set as well. Bank switching between these two sets allows for the varied graphics found throughout the game.

Graphic Codes 00 - 7F are common to all 80 - FF sets.

Hallway Graphics

Game	Sector 002C		
Byte #	Bytes	Colors set in Sector 4C, Bytes: 46, 50, 5A, 64 & 6E (Each byte sets color for 2 Floors)	Char-Codes
00000A80	00-07	L Bottom of Door: 4, 3, 2 & 1-pace away	C0
00000A88	08-0F	R Bottom of Door: 4, 3, 2 & 1-pace away	C1
00000A90	10-17	Blank	C2
00000A98	18-1F	Blank	C3
00000AA0	20-27	Blank	C4
00000AA8	28-2F	Blank	C5
00000AB0	30-37	Blank	C6
00000AB8	38-3F	Blank	C7

Large Block Graphics used when entering Rooms or Hallway Monster Appears

Game	Sector 002C		
Byte #	Bytes	Colors set in Sector 4C, Bytes: 47, 51, 5B, 65 & 6F (Each byte sets color for 2 Floors)	Char-Codes
00000AC0	40-47	(space char, (20) used when you enter an empty room)	C8
00000AC8	48-4F	0F0F 0F0F 0000 0000	C9
00000AD0	50-57	F0F0 F0F0 0000 0000	CA
00000AD8	58-5F	FFFF FFFF 0000 0000	CB
00000AE0	60-67	0000 0000 0F0F 0F0F	CC
00000AE8	68-6F	0F0F 0F0F 0F0F 0F0F	CD
00000AF0	70-77	F0F0 F0F0 0F0F 0F0F	CE
00000AF8	78-7F	FFFF FFFF 0F0F 0F0F	CF
00000B00	80-87	0000 0000 F0F0 F0F0	D0
00000B08	88-8F	0F0F 0F0F F0F0 F0F0	D1
00000B10	90-97	F0F0 F0F0 F0F0 F0F0	D2
00000B18	98-9F	FFFF FFFF F0F0 F0F0	D3
00000B20	A0-A7	0000 0000 FFFF FFFF	D4
00000B28	A8-AF	0F0F 0F0F FFFF FFFF	D5
00000B30	B0-B7	F0F0 F0F0 FFFF FFFF	D6
00000B38	B8-BF	FFFF FFFF FFFF FFFF	D7

Note: These large character blocks are used to create the (large) facsimile of the monster(s) seen in the room, as viewed when you first open the door. If no monsters are present, then a large graphic of an item in the room, e.g. a Vault or Stairway, is shown. If the room is completely empty then it is filled with the Space **Char-Code** – Hex 20 by **Char-Code** (C8).

Definable Hallway Graphic Space

Game	Sector 002C		
Byte #	Bytes	Color set in Sector 4C, Byte 8B (Color Byte is shared)	Char-Codes
00000B40	C0-C7	Blank	D8
00000B48	C8-CF	Blank	D9
00000B50	D0-D7	Blank	DA
00000B58	D8-DF	Blank	DB
00000B60	E0-E7	Blank	DC
00000B68	E8-EF	Blank	DD
00000B70	F0-F7	Blank	DE
00000B78	F8-FF	Blank	DF

Definable Hallway Graphic Space

Game	Sector 002D		
Byte #	Bytes	Color set in Sector 4C, Byte 8C (Color Byte is shared)	Char-Codes
00000B80	00-07	Blank	E0
00000B88	08-0F	Blank	E1
00000B90	10-17	Blank	E2
00000B98	18-1F	Blank	E3
00000BA0	20-27	Blank	E4
00000BA8	28-2F	Blank	E5
00000BB0	30-37	Blank	E6
00000BB8	38-3F	Blank	E7
		Color set in Sector 4C, Byte 8D (Color Byte is shared)	
00000BC0	40-47	Blank	E8
00000BC8	48-4F	Blank	E9
00000BD0	50-57	Blank	EA
00000BD8	58-5F	Blank	EB
00000BE0	60-67	Blank	EC
00000BE8	68-6F	Blank	ED
00000BF0	70-77	Blank	EE
00000BF8	78-7F	Blank	EF

Note: The Color Bytes assigned in Sector 004C: 8B, 8C and 8D, control the FG/ BG coloring of Graphic Banks 1 & 2 (Hallways and Room Contents) for their respective Char-Codes D8-EF.

Dynamic Game Graphic Workspace, Non-Definable (Sector 002D, Bytes 80-DF)

Game Byte #	Sector 002D Bytes	Color set by Sector 4C, Byte 8E (Color Byte is shared, but over written)	Char-Codes
00000C00	80-87	Reserved	F0
00000C08	88-8F	Reserved	F1
00000C10	90-97	Reserved	F2
00000C18	98-9F	Reserved	F3
00000C20	A0-A7	Reserved	F4
00000C28	A8-AF	Reserved	F5
00000C30	B0-B7	Reserved	F6
00000C38	B8-BF	Reserved	F7
		Color set by Sector 4C, Byte 8F (Color Byte is shared, but over written)	
00000C40	C0-C7	Reserved	F8
00000C48	C8-CF	Reserved	F9
00000C50	D0-D7	Reserved	FA
00000C58	D8-DF	Reserved	FB

Definable Game Graphic Workspace

Game Byte #	Sector 002D Bytes	Color set by Sector 4C, Byte 8F (Color Byte is shared)	
00000C60	E0-E7		FC
00000C68	E8-EF		FD
00000C70	F0-F7		FE
00000C78	F8-FF		FF

Total Program Bytes to current Section: BFF = 3,071 Bytes.

Notes on Bytes:

Note: **Char-Codes** F0 through FF are shared in common by both Hallway and Room Contents Graphic Banks (1 & 2). This includes both memory workspace and color control Bytes.

Sector 002D: In both 'Quest for the King' and 'Pennies', this sector contains volumes of Hex Digits. They differ from one another in Hex values, but are alike in that the values ***never*** change. These ***appear*** to be artifacts from the game 'Creation' in VDP memory. If someone learns otherwise, please advise!

80-9F	Monster Defense Graphic paged in, Fountain & Living Statue codes paged in when viewing “Hallway Fountains”. If a color besides '1E' is used, it will only persist until a key is pressed before reset by the module back to '1E' (Black on Red).	(F0-F3)
A0-BF	Duplicate Monster Defense Graphic, Stairs Up/ Down, Vault - all paged into workspace.	(F4-F7)
C0-DF	Monster Attack Graphic - paged in	(F8-FB)
E0-FF	Definable for use, however, if Color Code is changed from '6E', the new color will flicker briefly the first time the Monster Attack Graphic comes into play in each room.	(FC-FF)

Note: It is best not to use Bytes 80 - FF to avoid potential conflicts and undesired effects. If **Char-Codes** FC-FF are required for graphic purposes in a game, retain the Black on Red, Hex ‘1E’, color scheme.

Player (Characters) Saved Game Parameters (Stats for: "Continue Current Game" Option)

		Game	Sector 002E									
		Byte #	Bytes			Bytes			Bytes		Bytes	
Player Name	#1	00000C80	00-0E	#2	00000CBC	3C-4A	#3	00000CF8	78-86	#4	00000D34	B4-C2
Hit Points		00000C8F	0F		00000CCB	4B		00000D07	87		00000D43	C3
# Wounds		00000C90	10		00000CCC	4C		00000D08	88		00000D44	C4
Armor Type		00000C90	11		00000CCD	4D		00000D09	89		00000D45	C5
Armor Protection		00000C92	12		00000CCE	4E		00000D0A	8A		00000D46	C6
Shield Type		00000C93	13		00000CCF	4F		00000D0B	8B		00000D47	C7
Shield Protection		00000C94	14		00000CD0	50		00000D0C	8C		00000D48	C8
Weapon #1 Type		00000C95	15		00000CD1	51		00000D0D	8D		00000D49	C9
Weapon AV		00000C96	16		00000CD2	52		00000D0E	8E		00000D4A	CA
#Ammo if ranged		00000C97	17		00000CD3	53		00000D0F	8F		00000D4B	CB
Weapon #2 Type		00000C98	18		00000CD4	54		00000D10	90		00000D4C	CC
Weapon AV		00000C99	19		00000CD5	55		00000D11	91		00000D4D	CD
#Ammo if ranged		00000C9A	1A		00000CD6	56		00000D12	92		00000D4E	CE
Armor Bonus Points		00000C9B	1B		00000CD7	57		00000D13	93		00000D4F	CF
Weapon Bonus		00000C9C	1C		00000CD8	58		00000D14	94		00000D50	D0
Players Luck		00000C1D	1D		00000CD9	59		00000D15	95		00000D51	D1
Experience		00000C1E	1E-1F		00000CDA	5A-5B		00000D16	96-97		00000D52	D2-D3
Last Exp. Gain		00000C20	20-21		00000CDC	5C-5D		00000D18	98-99		00000D54	D4-D5
Player's Level		00000C22	22		00000CDE	5E		00000D1A	9A		00000D56	D6
Player's Class		00000C23	23		00000CDF	5F		00000D1B	9B		00000D57	D7
???		00000C24	24		00000CE0	60		00000D1C	9C		00000D58	D8
Player's Abilities		00000C25	25		00000CE1	61		00000D1D	9D		00000D59	D9
Last (room) Location		00000C26	26-27		00000CE2	62-63		00000D1E	9E-9F		00000D5A	DA-DB
Magical Item #1		00000C28	28		00000CE4	64		00000D20	A0		00000D5C	DC
#Remaining Uses		00000C29	29		00000CE5	65		00000D21	A1		00000D5D	DD
Magical Item #2		00000C2A	2A		00000CE6	66		00000D22	A2		00000D5E	DE
#Remaining Uses		00000C2B	2B		00000CE7	67		00000D23	A3		00000D5F	DF
Magical Item #3		00000C2C	2C		00000CE8	68		00000D24	A4		00000D60	E0
#Remaining Uses		00000C2D	2D		00000CE9	69		00000D25	A5		00000D61	E1
Magical Item #4		00000C2E	2E		00000CEA	6A		00000D26	A6		00000D62	E2
#Remaining Uses		00000C2F	2F		00000CEB	6B		00000D27	A7		00000D63	E3
Magical Item #5		00000C30	30		00000CEC	6C		00000D28	A8		00000D64	E4
#Remaining Uses		00000C31	31		00000CED	6D		00000D29	A9		00000D65	E5
Magical Item #6		00000C32	32		00000CEE	6E		00000D2A	AA		00000D66	E6
#Remaining Uses		00000C33	33		00000CEF	6F		00000D2B	AB		00000D67	E7
Magical Item #7		00000C34	34		00000CF0	70		00000D2C	AC		00000D68	E8
#Remaining Uses		00000C35	35		00000CF1	71		00000D2D	AD		00000D69	E9
Magical Item #8		00000C36	36		00000CF2	72		00000D2E	AE		00000D6A	EA
#Remaining Uses		00000C37	37		00000CF3	73		00000D2F	AF		00000D6B	EB
Magical Item #9		00000C38	38		00000CF4	74		00000D30	B0		00000D6C	EC
#Remaining Uses		00000C39	39		00000CF5	75		00000D31	B1		00000D6D	ED
Magical Item #10		00000C3A	3A		00000CF6	76		00000D32	B2		00000D6E	EE
#Remaining Uses		00000C3B	3B		00000CF7	77		00000D33	B3		00000D6F	EF

Party Saved Game Parameters

Bytes			
00000D70	F0-F7	(00)	Blank
00000D78	F8	(01)	Party location last time game was saved: 00=Top of Screen, 01=Bottom of screen
00000D79	F9	(73)	Party location last time game was saved. X-Y coordinates (
00000D7A	FA	(00)	Current Floor
00000D7B	FB	(02)	Players Speed, moves/ turn. Last needs to be a combat move
00000D7C	FC-FD	(000A)	Amount of Party Gold. Bytes FC=# of 4096 & 256 units, FD =# of 16 & 1 units
00000D7E	FE	(00)	Map Flag - # Maps you have
00000D7F	FF	(00)	Blank

Note: Bytes F8-F9 = same value as Sector 0050, Bytes 3B-3C for Party hallway locations.
Hex numbers in parentheses (xx) are the values found in "Quest for the king"

Party Saved Game Parameters (continued)

Game Byte #	Sector 002F Bytes	
00000D80	00	#Quest Items found
00000D81	01	#Quest Items remaining or destroyed
00000D82	02-03	Time left for Quest Item #1
00000D84	04-05	Time left for Quest Item #2
00000D86	06-07	Time left for Quest Item #3
00000D88	08-09	Time left for Quest Item #4
00000D8A	0A-0B	Time left for Quest Item #5
00000D8C	0C-0D	Time left for Quest Item #6
00000D8E	0E-0F	Time left for Quest Item #7
00000D90	10-11	Time left for Quest Item #8
00000D92	12	(03) Wandering Monster Probability (baseline 003A)
00000D93	13	(14) Amount of Rations remaining
00000D94	14	(00)
00000D95	15	(00)
00000D96	16	(00)
00000D97	17	(02) Ration Consumption Interval paces
00000D98	18	(00)
00000D99	19	(02) Party Healing Interval paces
00000D9A	1A	(00)
00000D9B	1B	(00)
00000D9C	1C	(00)
00000D9D	1D	(00)
00000D9E	1E	(00) Counter - ↑/↓ Combat Speed[1]
00000D9F	1F	(04) Default - ↑/↓ Combat Speed[2]
00000DA0	20	(00) Counter - ↑/↓ Wandering Monster Probability[1]
00000DA1	21	(04) Default - Wandering Monster Probability[2]
00000DA2	22	(00) Counter - ↑/↓ Consumption Interval[1]
00000DA3	23	(04) Default - Consumption Interval change[2]
00000DA4	24	(00) Counter - Healing Interval[1]
00000DA5	25	(04) Default - Healing Interval change[2]

Monster Combat Cue

Game Byte #	Sector 002F Bytes	
00000DA6	26-31	Monster Name
00000DB2	32	Level
00000DB3	33	DV
00000DB4	34	AV
00000DB5	35	Damage (Max Damage)
00000DB6	36	Special Attack Chance %
00000DB7	37	Special Attack Type (from list)
00000DB8	38	Special Attack Damage
00000DB9	39	Monster Sound Table Pointer
00000DBA	3A	Monster Graphic Pointer
00000DBB	3B	Negotiation (0 - 3)
00000DBC	3C	Mobility (1 - 4)
00000DBD	3D	Magical Resistance
00000DBE	3E	Last hex # of monster attributes is placed here?[3]
00000DBF	3F	Monster Speed
00000DC0	40	HP Monster #1
00000DC1	41	HP Monster #2
00000DC2	42	HP Monster #3
00000DC3	43	HP Monster #4
00000DC4	44	HP Monster #5
00000DC5	45	HP Monster #6
00000DC6	46	HP Monster #7
00000DC7	47	
00000DC8	48-49	Monster #1 Location
00000DCA	4A-4B	Monster #2 Location
00000DCC	4C-4D	Monster #3 Location
00000DCE	4E-4F	Monster #4 Location
00000DD0	50-51	Monster #5 Location
00000DD2	52-53	Monster #6 Location
00000DD4	54-55	Monster #7 Location
00000DD6	56-57	

Other Game Settings

00000DD8	58	(05)
00000DD9	59	(0A)
00000DDA	5A	(0F)
00000DDB	5B	(03)
00000DDC	5C	(0E) Lowering # decreases Party Combat Prob[4]
00000DDD	5D	(0A) Lowering # increases Party Combat Prob[5]
00000DDE	5E	(00)
00000DDF	5F	(00)
00000DE0	60	(1E) % Probability of Hearing Monster[6]
00000DE1	61	(28) % Prob. of not getting wounds at Vaults[7]
00000DE2	62-71	Word "Combination" - 16 bytes

Note: Monster's sound and graphic pointers are in reversed order in cue (compared to list) presumably because you can hear them before, when pressing 'L', or as you enter the room, before you see them.

[1] Paces/ key presses remaining (x 10) for change from baseline

[2] Change from baseline, used when the duration is not expressed in a spell or trap Measured in paces/ key presses, e.g. 04=40 key presses

[3] Possible Monster's Luck? Or nothing at all?

[4] All else being equal, possibly baseline Party Luck

[5] All else being equal, possibly baseline Monster Luck

[6] When pressing L for Listening at door

[7] i.e. when guessing wrong at Vault combination

Initial Values for Player Classes (Status) Defined (Class Stats with: "New Game" Option)

Sector	1ST CLASS		2ND CLASS		3RD CLASS		4TH CLASS	
002F	Game Byte #	(002F) Bytes	Game Byte #	(002F) Bytes	Game Byte #	(002F) Bytes	Game Byte #	(002F) Bytes
Class Name	00000DF2	72-7B	00000E08	88-91	00000E1E	9E-A7	00000E34	B4-BD
Gold to Start	00000DFC	7C	00000E12	92	00000E28	A8	00000E3E	BE
Class Abilities	00000DFD	7D	00000E13	93	00000E29	A9	00000E3F	BF
Armor (list)	00000DFE	7E	00000E14	94	00000E2A	AA	00000E40	C0
(00)	00000DFF	7F	00000E15	95	00000E2B	AB	00000E41	C1
Weapon #1 (list)	00000E00	80	00000E16	96	00000E2C	AC	00000E42	C2
(00)	00000E01	81	00000E17	97	00000E2D	AD	00000E43	C3
Weapon #2 (list)	00000E02	82	00000E18	98	00000E2E	AE	00000E44	C4
(00)	00000E03	83	00000E19	99	00000E2F	AF	00000E45	C5
Magical Item #1	00000E04	84	00000E1A	9A	00000E30	B0	00000E46	C6
(00)	00000E05	85	00000E1B	9B	00000E31	B1	00000E47	C7
Magical Item #2	00000E06	86	00000E1C	9C	00000E32	B2	00000E48	C8
(00)	00000E07	87	00000E1D	9D	00000E33	B3	00000E49	C9

Note: It would seem logical that **7F**=Shield Type, **81** & **83**=Ammo amount if a ranged weapon in **80** & **82**, but values entered here do not appear to do anything.

Graphics Associated with the Classes or Character Types

Game Byte #	Sector 002F Bytes	
00000E4A	CA-E9	1st Class - Defensive Graphic
00000E6A	EA-09	1st Class - Attack Graphic (to Sector 0030)
	Sector 0030	
00000E8A	0A-29	2nd Class - Defensive Graphic
00000EAA	2A-49	2nd Class - Attack Graphic
00000ECA	4A-69	3rd Class - Defensive Graphic
00000EEA	6A-89	3rd Class - Attack Graphic
00000F0A	8A-A9	4th Class - Defensive Graphic
00000F2A	AA-C9	4th Class - Attack Graphic

Characteristics of 56 Monster Types (Sectors 0030-0035)

#	Game Byte #	Name	Lev	DV	AV	Dam	SA%	SA Type	SA Dam	Graphic 1st hex	Sound 2nd hex	Mob / Neg -1st	Mag Res -2nd	Speed	?
1	00000F4A	CA-D5	D6	D7	D8	D9	DA	DB	DC	DD^1	DD^2	DE^1	DE^2	DF^1	DF^2
2	00000F60	E0-EB	EC	ED	EE	EF	F0	F1	F2	F3	F3	F4	F4	F5	F5
3	00000F76	F6-01	02	03	04	05	06	07	08	09	09	0A	0A	0B	0B
4	00000F8C	Sec 0031 0C-17	18	19	1A	1B	1C	1D	1E	1F	1F	20	20	21	21
5	00000FA2	22-2D	2E	2F	30	31	32	33	34	35	35	36	36	37	37
6	00000FB8	38-43	44	45	46	47	48	49	4A	4B	4B	4C	4C	4D	4D
7	00000FCE	4E-59	5A	5B	5C	5D	5E	5F	60	61	61	62	62	63	63
8	00000FE4	64-6F	70	71	72	73	74	75	76	77	77	78	78	79	79
9	00000FFA	7A-85	86	87	88	89	8A	8B	8C	8D	8D	8E	8E	8F	8F
10	00001010	90-9B	9C	9D	9E	9F	A0	A1	A2	A3	A3	A4	A4	A5	A5
11	00001026	A6-B1	B2	B3	B4	B5	B6	B7	B8	B9	B9	BA	BA	BB	BB
12	0000103C	BC-C7	C8	C9	CA	CB	CC	CD	CE	CF	CF	D0	D0	D1	D1
13	00001052	D2-DD	DE	DF	E0	E1	E2	E3	E4	E5	E5	E6	E6	E7	E7
14	00001068	E8-F3	F4	F5	F6	F7	F8	F9	FA	FB	FB	FC	FC	FD	FD
15	0000107E	Sec 0032 FE-09	0A	0B	0C	0D	0E	0F	10	11	11	12	12	13	13
16	00001094	14-1F	20	21	22	23	24	25	26	27	27	28	28	29	29
17	000010AA	2A-35	36	37	38	39	3A	3B	3C	3D	3D	3E	3E	3F	3F
18	000010C0	40-4B	4C	4D	4E	4F	50	51	52	53	53	54	54	55	55
19	000010D6	56-61	62	63	64	65	66	67	68	69	69	6A	6A	6B	6B
20	000010EC	6C-77	78	79	7A	7B	7C	7D	7E	7F	7F	80	80	81	81
21	00001102	82-8D	8E	8F	90	91	92	93	94	95	95	96	96	97	97
22	00001118	98-A3	A4	A5	A6	A7	A8	A9	AA	AB	AB	AC	AC	AD	AD
23	0000112E	AE-B9	BA	BB	BC	BD	BE	BF	C0	C1	C1	C2	C2	C3	C3
24	00001144	C4-CF	D0	D1	D2	D3	D4	D5	D6	D7	D7	D8	D8	D9	D9
25	0000115A	DA-E5	E6	E7	E8	E9	EA	EB	EC	ED	ED	EE	EE	EF	EF
26	00001170	F0-FB	FC	FD	FE	FF	00	01	02	03	03	04	04	05	05
27	00001186	Sec 0033 06-11	12	13	14	15	16	17	18	19	19	1A	1A	1B	1B
28	0000119C	1C-27	28	29	2A	2B	2C	2D	2E	2F	2F	30	30	31	31
29	000011B2	32-3D	3E	3F	40	41	42	43	44	45	45	46	46	47	47
30	000011C8	48-53	54	55	56	57	58	59	5A	5B	5B	5C	5C	5D	5D
31	000011DE	5E-69	6A	6B	6C	6D	6E	6F	70	71	71	72	72	73	73
32	000011F4	74-7F	80	81	82	83	84	85	86	87	87	88	88	89	89
33	0000120A	8A-95	96	97	98	99	9A	9B	9C	9D	9D	9E	9E	9F	9F
34	00001220	A0-AB	AC	AD	AE	AF	B0	B1	B2	B3	B3	B4	B4	B5	B5
35	00001236	B6-C1	C2	C3	C4	C5	C6	C7	C8	C9	C9	CA	CA	CB	CB
36	0000124C	CC-D7	D8	D9	DA	DB	DC	DD	DE	DF	DF	E0	E0	E1	E1
37	00001262	E2-ED	EE	EF	F0	F1	F2	F3	F4	F5	F5	F6	F6	F7	F7
38	00001278	Sec 0034 F8-03	04	05	06	07	08	09	0A	0B	0B	0C	0C	0D	0D
39	0000128E	0E-19	1A	1B	1C	1D	1E	1F	20	21	21	22	22	23	23
40	000012A4	24-2F	30	31	32	33	34	35	36	37	37	38	38	39	39
41	000012BA	3A-45	46	47	48	49	4A	4B	4C	4D	4D	4E	4E	4F	4F
42	000012D0	50-5B	5C	5D	5E	5F	60	61	62	63	63	64	64	65	65
43	000012E6	66-71	72	73	74	75	76	77	78	79	79	7A	7A	7B	7B
44	000012FC	7C-87	88	89	8A	8B	8C	8D	8E	8F	8F	90	90	91	91
45	00001312	92-9D	9E	9F	A0	A1	A2	A3	A4	A5	A5	A6	A6	A7	A7
46	00001328	A8-B3	B4	B5	B6	B7	B8	B9	BA	BB	BB	BC	BC	BD	BD
47	0000133E	BE-C9	CA	CB	CC	CD	CE	CF	D0	D1	D1	D2	D2	D3	D3
48	00001354	D4-DF	E0	E1	E2	E3	E4	E5	E6	E7	E7	E8	E8	E9	E9
49	0000136A	EA-F5	F6	F7	F8	F9	FA	FB	FC	FD	FD	FE	FE	FF	FF
50	00001380	Sec 0035 00-0B	0C	0D	0E	0F	10	11	12	13	13	14	14	15	15
51	00001396	16-21	22	23	24	25	26	27	28	29	29	2A	2A	2B	2B
52	000013AC	2C-37	38	39	3A	3B	3C	3D	3E	3F	3F	40	40	41	41
53	000013C2	42-4D	4E	4F	50	51	52	53	54	55	55	56	56	57	57
54	000013D8	58-63	64	65	66	67	68	69	6A	6B	6B	6C	6C	6D	6D
55	000013EE	6E-79	7A	7B	7C	7D	7E	7F	80	81	81	82	82	83	83
56	00001404	84-8F	90	91	92	93	94	95	96	97	97	98	98	99	99

Appendix IV - TOD Database Reference

List of Monster Special Attacks (20 Types Available)

		SA Name	SA			SA Name	SA
	Game	Sector 0035	Type		Game	Sector 0036	Type
#	Byte #	Bytes	Bytes	#	Byte #	Bytes	Bytes
01	0000141A	9A-A8	A9	0B	000014BA	3A-48	49
02	0000142A	AA-B8	B9	0C	000014CA	4A-58	59
03	0000143A	BA-C8	C9	0D	000014DA	5A-68	69
04	0000144A	CA-D8	D9	0E	000014EA	6A-78	79
05	0000145A	DA-E8	E9	0F	000014FA	7A-88	89
06	0000146A	EA-F8	F9	10	0000150A	8A-98	99
07	0000147A	FA-08	09 Sector 0036	11	0000151A	9A-A8	A9
08	0000148A	0A-18	19	12	0000152A	AA-B8	B9
09	0000149A	1A-28	29	13	0000153A	BA-C8	C9
0A	000014AA	2A-38	39	14	0000154A	CA-D8	D9

Monster Graphics (Monster Defense/ Attack Pose Graphics. The Graphic hex # found in the list of 56 Monsters refers to this List)

	Game	Sector 0036			Game	Sector 0038	
#	Byte #	Bytes	D/ A Pose		Byte #	Bytes	D/ A Pose
0	0000155A	DA-F9	Defense	8	0000175A	DA-F9	Defense
	0000157A	FA-19	A Sector 0037		0000177A	FA-19	A Sector 0039
1	0000159A	1A-39	Defense	9	0000179A	1A-39	Defense
	000015BA	3A-59	A		000017BA	3A-59	A
2	000015DA	5A-79	Defense	A	000017DA	5A-79	Defense
	000015FA	7A-99	A		000017FA	7A-99	A
3	0000161A	9A-B9	Defense	B	0000181A	9A-B9	Defense
	0000163A	BA-D9	A		0000183A	BA-D9	A
4	0000165A	DA-F9	Defense	C	0000185A	DA-F9	Defense
	0000167A	FA-19	A Sector 0038		0000187A	FA-19	A Sector 003A
5	0000169A	1A-39	Defense	D	0000189A	1A-39	Defense
	000016BA	3A-59	A		000018BA	3A-59	A
6	000016DA	5A-79	Defense	E	000018DA	5A-79	Defense
	000016FA	7A-99	A		000018FA	7A-99	A
7	0000171A	9A-B9	Defense	F	0000191A	9A-B9	Defense
	0000173A	BA-D9	A		0000193A	BA-D9	A

Global Game Settings for New/ Current Game

Game	Sector 003A		
Byte #	Bytes	Default/	Controls
0000195A	DA	(04)	Max # of Players (1 – 4)
0000195B	DB	(0A)	# of Rations/ unit purchased
0000195C	DC	(00)	(When a # is placed here, # of Rations never changes. Writes to Sector 002F, Byte 16.)
0000195D	DD	(02)	Baseline Ration Consumption Interval. 02 = 1 Ration consumed every 20 paces. Writes to Sector 2F, Byte 17.
0000195E	DE	(02)	Cost of Rations per Purchase. (x Factor)
0000195F	DF	(03)	Paces for Wandering Monster Probability check. Writes to Sector 002F, Byte 12.
00001960	E0	(02)	Baseline Party Healing Interval. Writes to Sector 002fF, Byte 19.
00001961	E1	(FC)	Number of Party Classes. (FF - FC or 1 - 4) Reverse notation = last Class only available in 1 Player Game.
00001962	E2	(0A)	Maximum # of Floors.
00001963	E3	(28)	\ E3 x E4 = # of Experience Points required to advance a Level. (E3 x 10 = for 1st Level)
00001964	E4	(0A)	/ (E4 = Increment Factor to advance to next Level)
00001965	E5	(14)	# of Ammo bought per purchase. (This entry will override any ammo quantity in the Ranged Weapon list if <E5.)
00001966	E6	(02)	# of Fountains or Statues in Hallways. (See Byte E9 to set as Fountain or Statue)
00001967	E7	(02)	# of Stairways down and up per Floor. See Note.
00001968	E8	(01)	Map Status: See Note.
00001969	E9	(07)	07="Who will use the Fountain?" 08="Who will use the Statue?" (Enables Hallway Statues.)
0000196A	EA	(03)	# of Players in the Current or a Saved Game. See Note.
0000196B	EB	(02)	# of Floors Selected in Game (also, default value if Restocking Dungeon or Continue Current Game)
0000196C	EC	(14)	# of Rooms/ Floor
0000196D	ED	(01)	Minimum # of Floors in Game
0000196E	EE	(03)	Max # of Vaults & Stores per Floor (total of both)
0000196F	EF	(0A)	Gold Factor for Statues
00001970	F0	(00)	
00001971	F1	(00)	
00001972	F2	(00)	
00001973	F3	(3C)	Probability of Hallway Monster attack after each Pace counter countdown is complete. (see Byte DF)
00001974	F4	(02)	# of Floors Selected in Game
00001975	F5	(02)	# of Floors Selected in Game
00001976	F6	(03)	# of Players in Current Game (probably for gold to start with calculation)
00001977	F7	(00)	Game Difficulty: 00=Easiest, 01=Medium, 02=Hardest*
00001978	F8	(00)	Current Floor
00001979	F9	(00)	
0000197A	FA	(00)	Current Player in Action Cue (01 - 04), 05 ends all individual player turns, 00=moving as a group
0000197B	FB	(01)	
0000197C	FC	(06)	Max amount of Healing that occurs when Rations are Consumed
0000197D	FD	(0A)	Cost of Healing factor x baseline (Purchased at Stores)
0000197E	FE	(1E)	
0000197F	FF	(06)	Max amount Player's HP can increase with new Level.

Global Game Settings of New/ Current Game (concluded)

Game	Sector 003B		
Byte #	Bytes	Default/	Controls
00001980	00	(06)	Your Location: 01=Hallway, 02=Rooms, 03=Hallway Fountain, 04= Stairs Down, 05=Stairs Up, 06=Store
00001981	01	(02)	If changed, resets back to'02'.
00001982	02	(00)	Direction the Party is facing: 00=North, 01=East, 02=South, 03=West
00001983	03	(00)	(60=monsters in room)
00001984	04	(00)	
00001985	05	(D6)	Total # of Bytes reserved for Rooms, Stairs & Hall Fountains
00001986	06	(00)	
00001987	07	(C8)	Total # Bytes reserved for Rooms (10 Bytes/ Room reserved)
00001988	08	(00)	
00001989	09	(CC)	Total # of Bytes reserved for Rooms & Stairs down
0000198A	0A	(00)	
0000198B	0B	(D0)	Total # of Bytes reserved for Rooms, Stairs down & Stairs up

Notes:

Byte E7 - **Stair graphics** appear if you can ascend or descend from Room, or if Room is actually flagged to be a stairway.

*Game Difficulty influences gold amount at start, wounds from guessing incorrectly at Vaults, # of Room Monsters & their Experience, etc.

Byte E8 - Maps: 00= not needed to descend, floor/ hallway always visible. 01=needed to descend, only explored floor/ hallway visible without map. >01 not needed to descend, only explored rooms are visible if returning to floor without a map (not hallways).

Byte EA - **Number of Players**. Used when: restarting a game/ restocking a dungeon, or if # of Players in a New Game is the same as in last game.

Note Sector 003B: Byte 00 appears to gives your Location for graphic purposes. 01= Hallway & use that set. 02= Room, all inside Room graphics & room contents graphics: vaults, chest, weapons, etc. except for Stairs (Rooms with Stairs have no other contents). 03= Hallway Fountain, etc.

Total Program Bytes to current Section: 190B = 6,411 Bytes

Rooms, Stairways, Hall Fountain locations and their Contents (Quest)

	Floor #1	Bytes	Floor #2	Bytes	Floor #3	Bytes	Floor #4	Bytes	Floor #5	Bytes
RM #1	Total Game Byte 0000198C	Sect. 003B 0C-15	Total Game Byte 00001A62	Sect. 003B E2-EB	Total Game Byte 00001B38	Sect. 003C B8-C1	Total Game Byte 00001C0E	Sect. 003D 8E-97	Total Game Byte 00001CE4	Sect. 003E 64-6D
RM #2	00001996	16-1F	00001A6C	EC-F5	00001B42	C2-CB	00001C18	98-A1	00001CEE	6E-77
RM #3	000019A0	20-29	00001A76	F6-FF	00001B4C	CC-D5	00001C22	A2-AB	00001CF8	78-81
RM #4	000019AA	2A-33	00001A80	Sect. 003C 00-09	00001B56	D6-DF	00001C2C	AC-B5	00001D02	82-8B
RM #5	000019B4	34-3D	00001A8A	0A-13	00001B60	E0-E9	00001C36	B6-BF	00001D0C	8C-95
RM #6	000019BE	3E-47	00001A94	14-1D	00001B6A	EA-F3	00001C40	C0-C9	00001D16	96-9F
RM #7	000019C8	48-51	00001A9E	1E-27	00001B74	F4-FD	00001C4A	CA-D3	00001D20	A0-A9
RM #8	000019D2	52-5B	00001AA8	28-31	00001B7E	FE-07	00001C54	D4-DD	00001D2A	AA-B3
RM #9	000019DC	5C-65	00001AB2	32-3B	00001B88	Sect. 003D 08-11	00001C5E	DE-E7	00001D34	B4-BD
RM #10	000019E6	66-6F	00001ABC	3C-45	00001B92	12-1B	00001C68	E8-F1	00001D3E	BE-C7
RM #11	000019F0	70-79	00001AC6	46-4F	00001B9C	1C-25	00001C72	F2-FB	00001D48	C8-D1
RM #12	000019FA	7A-83	00001AD0	50-59	00001BA6	26-2F	00001C7C	FC-05	00001D52	D2-DB
RM #13	00001A04	84-8D	00001ADA	5A-63	00001BB0	30-39	00001C86	Sect. 003E 06-0F	00001D5C	DC-E5
RM #14	00001A0E	8E-97	00001AE4	64-6D	00001BBA	3A-43	00001C90	10-19	00001D66	E6-EF
RM #15	00001A18	98-A1	00001AEE	6E-77	00001BC4	44-4D	00001C9A	1A-23	00001D70	F0-F9
RM #16	00001A22	A2-AB	00001AF8	78-81	00001BCE	4E-57	00001CA4	24-2D	00001D7A	FA-03
RM #17	00001A2C	AC-B5	00001B02	82-8B	00001BD8	58-61	00001CAE	2E-37	00001D84	Sect. 003F 04-0D
RM #18	00001A36	B6-BF	00001B0C	8C-95	00001BE2	62-6B	00001CB8	38-41	00001D8E	0E-17
RM #19	00001A40	C0-C9	00001B16	96-9F	00001BEC	6C-75	00001CC2	42-4B	00001D98	18-21
RM #20	00001A4A	CA-D3	00001B20	A0-A9	00001BF6	76-7F	00001CCC	4C-55	00001DA2	22-2B
Stairs Up	00001A54	D4-D7	00001B2A	AA-AD	00001C00	80-83	00001CD6	56-59	00001DAC	2C-2F
Stairs Down	00001A58	D8-DB	00001B2E	AE-B1	00001C04	84-87	00001CDA	5A-5D	00001DB0	30-33
Fountain #1	00001A5C	DC-DE	00001B32	B2-B4	00001C08	88-8A	00001CDE	5E-60	00001DB4	34-36
Fountain #2	00001A5F	DF-E1	00001B35	B5-B7	00001C0B	8B-8D	00001CE1	61-63	00001DB7	37-39
	Floor #6	Bytes	Floor #7	Bytes	Floor #8	Bytes	Floor #9	Bytes	Floor #10	Bytes
RM #1	00001DBA	3A-43	00001E90	10-19	00001F66	E6-EF	0000203C	BC-C5	00002112	92-9B
RM #2	00001DC4	44-4D	00001E9A	1A-23	00001F70	F0-F9	00002046	C6-CF	0000211C	9C-A5
RM #3	00001DCE	4E-57	00001EA4	24-2D	00001F7A	FA-03	00002050	D0-D9	00002126	A6-AF
RM #4	00001DD8	58-61	00001EAE	2E-37	00001F84	Sect. 0041 04-0D	0000205A	DA-E3	00002130	B0-B9
RM #5	00001DE2	62-6B	00001EB8	38-41	00001F8E	0E-17	00002064	E4-ED	0000213A	BA-C3
RM #6	00001DEC	6C-75	00001EC2	42-4B	00001F98	18-21	0000206E	EE-F7	00002144	C4-CD
RM #7	00001DF6	76-7F	00001ECC	4C-55	00001FA2	22-2B	00002078	F8-01	0000214E	CE-D7
RM #8	00001E00	80-89	00001ED6	56-5F	00001FAC	2C-35	00002082	Sect. 0042 02-0B	00002158	D8-E1
RM #9	00001E0A	8A-93	00001EE0	60-69	00001FB6	36-3F	0000208C	0C-15	00002162	E2-EB
RM #10	00001E14	94-9D	00001EEA	6A-73	00001FC0	40-49	00002096	16-1F	0000216C	EC-F5
RM #11	00001E1E	9E-A7	00001EF4	74-7D	00001FCA	4A-53	000020A0	20-29	00002176	F6-FF
RM #12	00001E28	A8-B1	00001EFE	7E-87	00001FD4	54-5D	000020AA	2A-33	00002180	Sect. 0043 00-09
RM #13	00001E32	B2-BB	00001F08	88-91	00001FDE	5E-67	000020B4	34-3D	0000218A	0A-13
RM #14	00001E3C	BC-C5	00001F12	92-9B	00001FE8	68-71	000020BE	3E-47	00002194	14-1D
RM #15	00001E46	C6-CF	00001F1C	9C-A5	00001FF2	72-7B	000020C8	48-51	0000219E	1E-27
RM #16	00001E50	D0-D9	00001F26	A6-AF	00001FFC	7C-85	000020D2	52-5B	000021A8	28-31
RM #17	00001E5A	DA-E3	00001F30	B0-B9	00002006	86-8F	000020DC	5C-65	000021B2	32-3B
RM #18	00001E64	E4-ED	00001F3A	BA-C3	00002010	90-99	000020E6	66-6F	000021BC	3C-45
RM #19	00001E6E	EE-F7	00001F44	C4-CD	0000201A	9A-A3	000020F0	70-79	000021C6	46-4F
RM #20	00001E78	F8-01	00001F4E	CE-D7	00002024	A4-AD	000020FA	7A-83	000021D0	50-59
Stairs Up	00001E82	Sect. 0040 02-05	00001F58	D8-DB	0000202E	AE-B1	00002104	84-87	000021DA	5A-5D
Stairs Down	00001E86	06-09	00001F5C	DC-DF	00002032	B2-B5	0000108	88-8B	000021DE	5E-61
Fountain #1	00001E8A	0A-0C	00001F60	E0-E2	00002036	B6-B8	0000210C	8C-8E	000021E2	62-64
Fountain #2	00001E8D	0D-0F	00001F63	E3-E5	00002039	B9-BB	0000210F	8F-91	000021E5	65-67

Note: Dungeons have a **maximum of 2,140 Bytes** available for their design.
Each room requires 10 bytes for storing information on location & contents.
Each Stairway Down requires 2 bytes of storage. Each Stairway Up requires 2 bytes of storage.
Each Hallway Fountain requires 3 bytes of storage. However, note the following:
For an even number of Fountains per Floor:
Bytes = Fountains x 3
For an odd numbers of Fountains per Floor:
Bytes = (Fountains x 3) +1

The sum total of bytes (per floor) must be computed and then multiplied by the intended maximum number of floors for the game. This result should **total 2,140 Bytes** or less.

Miscellaneous Global Settings - **Sector 0043** (continued)

Byte #	**Bytes**	
000021E8	**68-76**	Monster selection by Exp Class per floor: (Easy = 68–6C; Med. = 6D–71; Hard = 72–76)
000021F7	**77**	(00)
000021F8	**78-93**	YOU FOUND THE COMBINATION! (28 Bytes available for phrase)
00002214	**94-A3**	GENERAL STORE (16 Bytes available for phrase)
00002224	**A4-B3**	Pointers to the 16 Monster Sounds available in the TOD Module. The Sound Nibble found in the list of 56 Monsters refers to this List, which in turn cues the TOD Module for the desired sound effect. ***Default:***: 00 to 0F.
00002234	**B4-B5**	(C7F2)
00002236	**B6-EE**	(00) Unused?
0000226F	**EF**	(00) See Note:
00002270	**F0**	(FF) Resets back to "FF" if value is changed.
00002271	**F1**	(03)
00002272	**F2-F3**	(00)
00002274	**F4**	(1E)
00002275	**F5**	(B8)
00002276	**F6-F7**	(0173) Last Map Hallway location (but not Rooms except for Ground Floor) Same as 002E, bytes F7-F8 with New Game.
00002278	**F8-FC**	(00)
0000227D	**FD**	(03) Resets itself back to "03" if value is changed.

Note: Byte # 000043EF Sector 0043, Byte **EF:** 00-70 Store on Ground Floor: 00–30 Can use Fountains if Dead, Food Heals, 40–70 Player cannot use Fountains if Dead, Food does not Heal. 80-F0 No Store on ground Floor: 80–B0 Can use Fountain if Dead, Food Heals, C0–F0 Cannot use Fountains if Dead, Food does not Heal.

Vault & Store Settings

Game	**Sector 0043**		**Game**	**Sector 0044**	
Byte #	**Bytes**		**Byte #**	**Bytes**	
0000227E	**FE-02**	1st Floor	00002297	**17-1B**	6th Floor
00002283	**03-07**	2nd Floor **Sector 0044**	0000229C	**1C-20**	7th Floor
00002288	**08-0C**	3rd Floor	000022A1	**21-25**	8th Floor
0000228D	**0D-11**	4th Floor	000022A6	**26-2A**	9th Floor
00002292	**12-16**	5th Floor	000022AB	**2B-2F**	10th Floor

Note: Five Bytes are reserved for up to 5 Stores and/ or Vaults per Floor. If Byte contains '01' = Store, '02' = Vault. To be functional the corresponding Vault Combination ranges need to be set and Byte 'EE' in Sector 003A needs to be set correctly.

Vault Combination Settings

Game	**Sector 0044**		**Game**	**Sector 0044**	
Byte #	**Bytes**		**Byte #**	**Bytes**	
000022B0	**30-34**	1st Floor	000022C9	**49-4D**	6th Floor
000022B5	**35-39**	2nd Floor	000022CE	**4E-52**	7th Floor
000022BA	**3A-3E**	3rd Floor	000022D3	**53-57**	8th Floor
000022BF	**3F-43**	4th Floor	000022D8	**58-5C**	9th Floor
000022C4	**44-48**	5th Floor	000022DD	**5D-61**	10th Floor

Note: Five Bytes are reserved per floor (for up to 5 Vaults). Bytes correspond to specific Vaults listed above. E.g. Byte 30 sets the Combination range for Vault in Byte 'FE' of Sector 0043, etc. For each Byte 1st digit = length (#) of combination, 2nd digit = range of numbers. For example a '34' = Combination of 3 numbers, ranging from 1 - 4. If a Store '01' is present, then leave the corresponding byte at '00'. Unused Combination spaces can be seeded for items to prevent too many empty rooms.

Fountain Settings - Sector 0044 (continued)

Game Byte #	Sector 0044 Bytes	
000022E2	**62-75**	Hallway Fountains are assigned an initial value of 0-3, which refer to one of 4 Banks of effect probabilities. These values are set here, 1 byte/ Fountain. There is space for 20 Fountain flags. Fountains in excess of 20 are assigned a "0" value. See **Sector 004A**, Bytes E4-15.
000022F6	**76-79**	# of Players & their order, i.e. 01, 02, 03, 04. This changes when you press "O" & change Party Order.
000022FA	**7A-83**	Max. # of Room Monsters by Floors: 1 Byte/ Floor (7A=Floor 1, 7B=2, 7C=3, etc.) (To this is added # Players, difficulty level, etc.) To set <u>all Rooms</u> on a floor with a set # of Monsters/ Room, use: FC=7, FB=6, FA=5, F9=4, F8=3, F7=2, F6=1, F5=0
00002304	**84-8F**	Dynamic workspace (snapshot of saved) that represents room computations, contents, location of vault, chest, stairs, etc. Always '00' for halls
00002310	**90-93**	Initial Hit Points for the 4 Classes of Characters

Note: 0D0F in Bytes **84-85** = "stairs are present", 0D05 (repeating x 5) in bytes **86-8F** ="Vault in room". As dynamic values, these cannot be edited and saved.

Floor Information - *10 Bytes of data/ Floor to set limits on the Quantities of various Room Items*

Game Byte #	Sector 0044 Bytes		Game Byte #	Sector 0044 Bytes	
00002314	**94-9D**	Floor 1 Information	00002346	**C6-CF**	Floor 6 Information
0000231E	**9E-A7**	Floor 2 Information	00002350	**D0-D9**	Floor 7 Information
00002328	**A8-B1**	Floor 3 Information	0000235A	**DA-E3**	Floor 8 Information
00002332	**B2-BB**	Floor 4 Information	00002364	**E4-ED**	Floor 9 Information
0000233C	**BC-C5**	Floor 5 Information	0000236E	**EE-F7**	Floor 10 Information

Lists of Available Party Weapons, Armor & Shields (Sectors 0044 - 0047)

Hand Weapons List

<u>#</u>	Game Byte #	Name Bytes	Attribute Bytes
1	00002378	**F8-06**	**07-09** **Sector 0045**
2	0000238A	**0A-18**	**19-1B**
3	0000238C	**1C-2A**	**2B-2D**
4	000023AE	**2E-3C**	**3D-3F**
5	000023C0	**40-4E**	**4F-51**
6	000023D2	**52-60**	**61-63**
7	000023E4	**64-72**	**73-75**
8	000023F6	**76-84**	**85-87**

Ranged Weapons List

<u>#</u>		Name	Attributes	Ammo Name
9	00002408	**88-96**	**97-9C**	**9D-A9**
A	0000242A	**AA-B8**	**B9-BE**	**BF-CB**
B	0000244C	**CC-DA**	**DB-E0**	**E1-ED**
			Sector 0046	
C	0000246E	**EE-FC**	**FD-02**	**03-0F**
D	00002490	**10-1E**	**1F-24**	**25-31**
E	000024B2	**32-40**	**41-46**	**47-53**
F	000024D4	**54-62**	**63-68**	**69-75**
10	000024F6	**76-84**	**85-8A**	**8B-97**

Armor List

#	Game Byte #	Name Bytes	Attribute Bytes
1	00002518	**98-A6**	**A7-A9**
2	0000252A	**AA-B8**	**B9-BB**
3	0000253C	**BC-CA**	**CB-CD**
4	0000254E	**CE-DC**	**DD-DF**
5	00002560	**E0-EE**	**EF-F1**
		Sector 0047	
6	00002572	**F2-00**	**01-03**
7	00002584	**04-12**	**13-15**
8	00002596	**16-24**	**25-27**

Shield List

#		Name	Attributes
9	000025A8	**28-36**	**37-39**
A	000025BA	**3A-48**	**49-4B**
B	000025CC	**4C-5A**	**5B-5D**
C	000025DE	**5E-6C**	**6D-6F**
D	000025F0	**70-7E**	**7F-81**
E	00002602	**82-90**	**91-93**

Note: Weapon, Armor & Shield names can be up to 15 characters long, Ammo names up to 13 characters. See Lists.

Appendix IV - TOD Database Reference

Lists of Magical Item Categories (Sector 0047)

	Game Byte #	Name Bytes	Attribute Bytes		Game Byte #	Name Bytes	Attribute Bytes	
Category #1	00002614	94-9E	9F-A3	**Category #5**	00002654	D4-DE	DF-E3	
Category #2	00002624	A4-AE	AF-B3	**Category #6**	00002664	E4-EE	EF-F3	
Category #3	00002634	B4-BE	BF-C3	**Category #7**	00002674	F4-FE	FF-03	**Sector 0048**
Category #4	00002644	C4-CE	CF-D3	**Category #8**	00002684	04-0E	0F-13	

Note: Up to 8 Categories of Magical Items are available in TOD. Each Category is assigned 5 specific "Spells" or Items. Category Names can be repeated, e.g. Quest has 3 Categories named “Scroll”, however each spell is usually given a specific name. The maximum # of spells or magical Items is 40.

Category Names can be up to 11 characters long. **Attributes**: the first 4 attribute bytes contains the 4 Char-codes for the Category’s Graphic representation. E.g. a Scroll or a Lantern. The last attribute byte (5th) is the Dungeon Level the item can first be found. If the Dungeon Level is written in Reverse Notation, then only a “Wizard Class” Player can use the 5 Spells of that Category. Any Category can be reserved for “Wizards Only”. (A Wizard Class Player has hex 1 as the first hex digit of his Class attribute byte.)

List of 40 Magical Items/ Effects

	Game	Name	Attribute		Game	Name	Attribute
Category #1	Byte #	Bytes	Bytes	**Category #5**	Byte #	Bytes	Bytes
Spell #1	00002694	Sector 0048 14-22	23-25	**Spell #21**	000027FC	7C-8A	8B-8D
Spell #2	000026A6	26-34	35-37	**Spell #22**	0000280E	8E-9C	9D-9F
Spell #3	000026B8	38-46	47-49	**Spell #23**	00002820	A0-AE	AF-B1
Spell #4	000026CA	4A-58	59-5B	**Spell #24**	00002832	B2-C0	C1-C3
Spell #5	000026DC	5C-6A	6B-6D	**Spell #25**	00002844	C4-D2	D3-D5
Category #2				**Category #6**			
Spell #6	000026EE	6E-7C	7D-7F	**Spell #26**	00002856	D6-E4	E5-E7
Spell #7	00002700	80-8E	8F-91	**Spell #27**	00002868	E8-F6	F7-F9
Spell #8	00002712	92-A0	A1-A3	**Spell #28**	0000287A	FA-08	Sector 004A 09-0B
Spell #9	00002724	A4-B2	B3-B5	**Spell #29**	0000288C	0C-1A	1B-1D
Spell #10	00002736	B6-C4	C5-C7	**Spell #30**	0000289E	1E-2C	2D-2F
Category #3				**Category #7**			
Spell #11	00002748	C8-D6	D7-D9	**Spell #31**	000028B0	30-3E	3F-41
Spell #12	0000275A	DA-E8	E9-EB	**Spell #32**	000028C2	42-50	51-53
Spell #13	0000276C	EC-FA	FB-FD	**Spell #33**	000028D4	54-62	63-65
Spell #14	0000277E	FE-0C	Sector 0049 0D-0F	**Spell #34**	000028E6	66-74	75-77
Spell #15	00002790	10-1E	1F-21	**Spell #35**	000028F8	78-86	87-89
Category #4				**Category #8**			
Spell #16	000027A2	22-30	31-33	**Spell #36**	0000290A	8A-98	99-9B
Spell #17	000027B4	34-42	43-45	**Spell #37**	0000291C	9C-AA	AB-AD
Spell #18	000027C6	46-54	55-57	**Spell #38**	0000292E	AE-BC	BD-BF
Spell #19	000027D8	58-66	67-69	**Spell #39**	00002940	C0-CE	CF-D1
Spell #20	000027EA	6A-78	79-7B	**Spell #40**	00002952	D2-E0	E1-E3

Note: Magical Item Names can be 15 characters in length. **Attributes**: 1st Byte = Spell #, from the “List of Known Spells”. 2nd Byte = # of uses **OR** duration in paces. If 2nd Byte = '00' then the effect takes place i*mmediately* upon contact. 3rd Byte = maximum effect or impact.

Fountain Effects & Probabilities

Game Byte #	Sector 004A Bytes	
00002964	**E4-ED**	Fountain Effects (each effect = 1 Byte, room for 10 effects)
0000296E	**EE-F7**	1st Probability Bank
	Sector 004B	
00002978	**F8-01**	2nd Probability Bank
00002982	**02-0B**	3rd Probability Bank
0000298C	**0C-15**	4th Probability Bank

Note: The number of effects must correspond to the number of probabilities. I.e. if you are only using 8 effects, then only use 8 bytes for probabilities and leave the last 2 bytes '0000', 6 effects, then only 6 bytes of probabilities, etc. The probabilities should be incremental in each Bank and the last one should be '64' = 100% Probability to prevent unwanted effects in game performance.

Quest Objects - Sector 004B

	Game Byte #	Name Bytes	Attribute Bytes		Game Byte #	Name Bytes	Attribute Bytes
Item #1	00002996	**16-20**	**21-28**	**Item #5**	000029E2	**62-6C**	**6D-74**
Item #2	000029A9	**29-33**	**34-3B**	**Item #6**	000029F5	**75-7F**	**80-87**
Item #3	000029BC	**3C-46**	**47-4E**	**Item #7**	00002A08	**88-92**	**93-9A**
Item #4	000029CF	**4F-59**	**5A-61**	**Item #8**	00002A1B	**9B-A5**	**A6-AD**

Note: Quest Item names can be up to 11 characters long. **Attribute Bytes**: 1st Byte = Floor where Item may first be found. 2nd Byte = Last Floor Item can be found. (If fewer Floors are selected for play then the Item is placed on the lowest Floor) 3rd through 6th Bytes = Graphic pointers. Bytes = 7 & 8 = Time factor to find Item (Factor x # Floors selected = count down # to find specific Item.

Ranged & Magical Attack Sequences

Game Byte #	Sector 004B Bytes	
00002A2E	AE-B5	**Char-Codes for Projectile Weapon Sequence** (AE - B1=Attack Sequence, B2=Impact Effect, B3=Effect offset, B4=Initial Effect, B5=Ending Graphic e.g. 20 or Space)
00002A36	B6-BD	**Char-Codes for Magical Weapon Sequence** (B6 - B9=Attack Sequence, BA=Impact Effect, BB=Effect offset, BC=Initial Effect, BD=Ending Graphic e.g. 20 or Space)

Note: The effects do not replace, but cause an overlay effect about the graphic. I.E. when a ranged or magical weapon is discharged, and when it successfully hits its target.

Chest Traps & Effects

Game Byte #	Sector 004B Bytes	
00002A33	BE-C5	The word: "Open" (when Opening a Chest)

List of 10 Types of Chest Traps & their Effects

	Game Byte #	Name Bytes	Effect Bytes		Game Byte #	Name Bytes	Effect Bytes
Trap #1	00002A46	**C6-CF**	**D0-D1**	**Trap #6**	00002A82	**02-0B**	**0C-0D**
Trap #2	00002A52	**D2-DB**	**DC-DD**	**Trap #7**	00002A8E	**0E-17**	**18-19**
Trap #3	00002A5E	**DE-E7**	**E8-E9**	**Trap #8**	00002A9A	**1A-23**	**24-25**
Trap #4	00002A6A	**EA-F3**	**F4-F5**	**Trap #9**	00002AA6	**26-2F**	**30-31**
Trap #5	00002A76	**F6-FF**	**00-01** (Sector 004C)	**Trap #10**	00002AB2	**32-3B**	**3C-3D**

Note: Trap names can be 10 characters long. **Effects**: 1st Byte = Spell used, 2nd Byte = probability of this Spell or Effect occurring. The probabilities should be an incremental progression and the last Trap must = 64 or 100% probability. This Trap List determines the effects for all booby-trapped Chests.

***Color Codes* Note**: (The two hex digits of each byte control the *Foreground/ Background* colors as per TI Basic and XB.)

0	Transparent	4	Dark Blue	8	Medium Red	C	Dark Green
1	Black	5	Light Blue	9	Light Red	D	Magenta
2	Medium Green	6	Dark Red	A	Dark Yellow	E	Gray
3	Light Green	7	Cyan	B	Light Yellow	F	White

Color Code Tables for Graphic Character Sets (FG/ BG) - Sector 004C

Table #1 - Hallway Color Sets - by Floor #: (Bytes express FG/BG Colors)

Floors = 1&2		3&4		5&6		7&8		9&10		
Game		Game		Game		Game		Game		Colors for:
Byte #	Byte	Byte #	Byte	Byte #	Byte	Byte #	Byte	Byte #	Byte	Char-Codes
00002ABE	3E	00002AC8	48	00002AD2	52	00002ADC	5C	00002AE6	66	80-87
00002ABF	3F	00002AC9	49	00002AD3	53	00002ADD	5D	00002AE7	67	88-8F
00002AC0	40	00002ACA	4A	00002AD4	54	00002ADE	5E	00002AE8	68	90-97
00002AC1	41	00002ACB	4B	00002AD5	55	00002ADF	5F	00002AE9	69	98-9F
00002AC2	42	00002ACC	4C	00002AD6	56	00002AE0	60	00002AEA	6A	A0-A7
00002AC3	43	00002ACD	4D	00002AD7	57	00002AE1	61	00002AEB	6B	A8-AF
00002AC4	44	00002ACE	4E	00002AD8	58	00002AE2	62	00002AEC	6C	B0-B7
00002AC5	45	00002ACF	4F	00002AD9	59	00002AE3	63	00002AED	6D	B8-BF
00002AC6	46	00002AD0	50	00002ADA	5A	00002AE4	64	00002AEE	6E	C0-C7
00002AC7	47	00002AD1	51	00002ADB	5B	00002AE5	65	00002AEF	6F	C8-D7

Char-Codes	Description
80-87	R & L walls, but not directly ahead or wall/ ceiling interface. Floor.
88-8F	Ceiling. R & L wall/ ceiling interface.
90-97	Wall viewed straight on. Door/ wall 4 paces away & straight on 1st door view.
98-9F	Door/ wall interface when viewed from side.
A0-A7	Door - 4,3,2 & 1 paces away side view, 3,2 &1 directly ahead. Doorknob.
A8-AF	Wall/ ceiling interface viewed straight on, 2 paces away.
B0-B7	Hallway Fountains.
B8-BF	Hallway Fountains.
C0-C7	Floor/ door interface when viewed straight on & 1 pace away head-on.
C8-D7	Inside of Room, Steps, Vault as seen when door 1st opens (from outside) (2 Sets).

Table #2 - Room Contents Color Table (Bytes express FG/BG Colors)

Player Characters Classes 1 - 4, Regular ASCII Characters 32 - 127, Rooms and all Room Contents

Game Byte #	Byte	Char-Codes	Description	Sector Found
00002AF0	70	00-07	Character Class #1 D/ A	**Sector 0026**
00002AF1	71	08-0F	Character Class #2 D/ A	
00002AF2	72	10-17	Character Class #3 D/ A	
00002AF3	73	18-1F	Character Class #4 D/ A	
00002AF4	74	20-27	ASCII Codes 32-39 **Sector 0027**	
00002AF5	75	28-2B	ASCII Codes 40-47	
00002AF6	76	30-37	ASCII Codes 48-55	
00002AF7	77	38-3F	ASCII Codes 56-63	
00002AF8	78	40-47	ASCII Codes 64-71 **Sector 0028**	
00002AF9	79	48-4F	ASCII Codes 72-79	
00002AFA	7A	50-57	ASCII Codes 80-87	
00002AFB	7B	58-5F	ASCII Codes 88-95	
00002AFC	7C	60-67	ASCII Codes 96-103 **Sector 0029**	
00002AFD	7D	68-6F	ASCII Codes 104-111	
00002AFE	7E	70-77	ASCII Codes 112-119	
00002AFF	7F	78-7F	ASCII Codes 120-127	
00002B00	80	80-87	(Sector 004C) Bytes: 90-CF	**Sector 004C**
00002B01	81	88-8F	Bytes: D0-(Sector 004D) 0F	**Sector 004D** (Byte 81 does not appear to work?)
00002B02	82	90-97	Sector 004D Bytes: 10-4F	(Byte 82 works on Ground Floor only - Game Logo)
00002B03	83	98-9F	Bytes: 50-8F	
00002B04	84	A0-A7	Bytes: 90-CF	
00002B05	85	A8-AF	Bytes: D0-(Sector 004E) 0F	**Sector 004E**
00002B06	86	B0-B7	Sector 004E Bytes: 10-4F	
00002B07	87	B8-BF	Bytes: 50-8F	
00002B08	88	C0-C7	Bytes: 90-CF	
00002B09	89	C8-CF	Bytes: D0-(Sector 004F) 0F	**Sector 004F**
00002B0A	8A	D0-D7	Sector 004F Bytes: 10-4F	
00002B0B	8B	D8-DF	Bytes: 50-8F	(shared with Hallway from here to Char-Code 'FF')
00002B0C	8C	E0-E7	Bytes: 90-CF	
00002B0D	8D	E8-EF	Bytes: D0-(Sector 0050) 0F	**Sector 0050**
00002B0E	8E	F0-F7	Bytes: 80-BF (Sector 002D)	Changes color of Stairs & Monsters-D (Monsters initially only
00002B0F	8F	F8-FF	Bytes: C0-FF	Monsters-A

Note: Color Codes for Char-Codes D8 - FF in the Hallway graphics bank are shared with and set by their counterparts in the Rooms and Contents graphics bank

Graphic Character Space for Rooms & their Contents – Graphics Bank #2 (>7F Char-Codes)

The following graphics do not appear to be re-locatable: ***Stairs, Game Logo, Vault & Room Graphics***

```
Game          Sector 004C
Byte #        Bytes
              90-CF   Definable Graphic space, Color set by Byte 80
00002B10      90-97   Char-Code 80
00002B18      98-9F   Char-Code 81
00002B20      A0-A7   Char-Code 82
00002B28      A8-AF   Char-Code 83
00002B30      B0-B7   Char-Code 84
00002B38      B8-BF   Char-Code 85
00002B40      C0-C7   Char-Code 86
00002B48      C8-CF   Char-Code 87
00002B50      D0-EF   Char-Codes 88-8B    Stairs Up Graphic , Color  set by Byte 8E (Cannot be relocated)
00002B70      F0-0F   Char-Codes 8C-8F    Sector 004D) Stairs Down Graphic  (Cannot be relocated)
              Sector 004D - Graphic Character Space for Room Contents
00002B90      10-2F   Char-Codes 90-93 Game Logo Graphic, Color set by Byte 82          (1)
00002BB0      30-4F   Char-Codes 94-97 Vault Graphics, Color set by Byte 82    (2)
              50-8F   Definable Graphic space, Color set by Byte 83
00002BD0      50-57   Char-Code 98\
00002BD8      58-5F   Char-Code 99 \                                          (3)
00002BE0      60-67   Char-Code 9A /
00002BE8      68-6F   Char-Code 9B/
00002BF0      70-77   Char-Code 9C\
00002BF8      78-7F   Char-Code 9D \                                          (4) etc.
00002C00      80-87   Char-Code 9E /
00002C08      88-8F   Char-Code 9F/
              90-CF   Definable Graphic space, Color set by Byte 84
00002C10      90-97   Char-Code A0
00002C18      98-9F   Char-Code A1
00002C20      A0-A7   Char-Code A2
00002C2B      A8-AF   Char-Code A3
00002C30      B0-B7   Char-Code A4
00002C38      B8-BF   Char-Code A5
00002C40      C0-C7   Char-Code A6
00002C48      C8-CF   Char-Code A7
              D0-0F   Definable Graphic space, Color set by Byte 85
00002C50      D0-D7   Char-Code A8
00002C58      D8-DF   Char-Code A9
00002C60      E0-E7   Char-Code AA
00002C68      E8-EF   Char-Code AB
00002C70      F0-F7   Char-Code AC
00002C78      F8-FF   Char-Code AD
00002C80      00-07   Char-Code AE   (Sector 004E)
00002C88      08-0F   Char-Code AF
              Sector 004E - Graphic Character Space for Room Contents
              10-4F   Definable Graphic space, Color set by Byte 86
00002C90      10-17   Char-Code B0
00002C98      18-1F   Char-Code B1
00002CA0      20-27   Char-Code B2
00002CA8      28-2F   Char-Code B3
00002CB0      30-37   Char-Code B4
00002CB8      38-3F   Char-Code B5
00002CC0      40-47   Char-Code B6
00002CC8      48-4F   Char-Code B74E
              50-8F   Definable Graphic space, Color set by Byte 87
00002CD0      50-57   Char-Code B8
00002CD8      58-5F   Char-Code B9
00002CE0      60-67   Char-Code BA
00002CE8      68-6F   Char-Code BB
00002CF0      70-77   Char-Code BC
00002CF8      78-7F   Char-Code BD
00002D00      80-87   Char-Code BE
00002D08      88-8F   Char-Code BF
```

Game	**Sector 004E** (cont) - ***Graphic Character Space for Room Contents***		
Byte #	90-CF	Definable Graphic space, Color set by Byte 88	
00002D10	**90-97**	Char-Code C0	
00002D18	**98-9F**	Char-Code C1	
00002D20	**A0-A7**	Char-Code C2	
00002D28	**A8-AF**	Char-Code C3	
00002D30	**B0-B7**	Char-Code C4	
00002D38	**B8-BF**	Char-Code C5	
00002D40	**C0-C7**	Char-Code C6	
00002D48	**C8-CF**	Char-Code C7	
	D0-0F	Definable Graphic space, Color set by Byte 89	
00002D50	**D0-D7**	Char-Code C8	
00002D58	**D8-DF**	Char-Code C9	
00002D60	**E0-E7**	Char-Code CA	
00002D68	**E8-EF**	Char-Code CB	
00002D70	**F0-F7**	Char-Code CC	
00002D78	**F8-FF**	Char-Code CD	
	Sector 004F - ***Graphic Character Space for Room Contents***		
00002D80	**00-07**	Char-Code CE	
00002D88	**08-0F**	Char-Code CF	
	10-4F	Definable Graphic space, Color set by Byte 8A	
00002D90	**10-17**	Char-Code D0	
00002D98	**18-1F**	Char-Code D1	
00002DA0	**20-27**	Char-Code D2	
00002DA8	**28-2F**	Char-Code D3	
00002DB0	**30-37**	Char-Code D4	
00002DB8	**38-3F**	Char-Code D5	
00002DC0	**40-47**	Char-Code D6	
00002DC8	**48-4F**	Char-Code D7	
	50-8F	**Room Graphics**, Color set by Byte 8B (Cannot be relocated)	
00002DD0	**50-57**	Char-Code D8	Vertical Room Wall
00002DD8	**58-5F**	Char-Code D9	Horizontal Room Wall
00002DE0	**60-67**	Char-Code DA	Vertical Room Doorway
00002DE8	**68-6F**	Char-Code DB	Horizontal Room Doorway
00002DF0	**70-77**	Char-Code DC	Room Corner Post
00002DF8	**78-7F**	Char-Code DD	Grid Pattern outside a Room
00002E00	**80-87**	Char-Code DE	Horizontal Room Jct. (not actually used - free)
00002E08	**88-8F**	Char-Code DF	Vertical Room Jct. (not actually used - free)
	90-CF	Definable Graphic space, Color set by Byte 8C	
00002E10	**90-97**	Char-Code E0	
00002E18	**98-9F**	Char-Code E1	
00002E20	**A0-A7**	Char-Code E2	
00002E28	**A8-AF**	Char-Code E3	
00002E30	**B0-B7**	Char-Code E4	
00002E38	**B8-BF**	Char-Code E5	
00002E40	**C0-C7**	Char-Code E6	
00002E48	**C8-CF**	Char-Code E7	
	D0-FF	Definable Graphic space, Color set by Byte 8D	
00002E50	**D0-D7**	Char-Code E8	
00002E58	**D8-DF**	Char-Code E9	
00002E60	**E0-E7**	Char-Code EA	
00002E68	**ED-EF**	Char-Code EB	
00002E70	**F0-F7**	Char-Code EC	
00002E78	**F8-FF**	Char-Code ED	
	Sector 0050 - ***Graphic Character Space for Room Contents***		
00002E80	**00-07**	Char-Code EE	
00002E88	**08-0F**	Char-Code EF	

***Color Codes* Note**: (The two hex digits of each byte control the *Foreground/ Background* colors as per TI Basic and XB.)

0 Transparent	4 Dark Blue	8 Medium Red	C Dark Green
1 Black	5 Light Blue	9 Light Red	D Magenta
2 Medium Green	6 Dark Red	A Dark Yellow	E Gray
3 Light Green	7 Cyan	B Light Yellow	F White

Char-Code Location Pointers for Room Contents

Game	Sector 0050	
Byte #	Bytes	
00002E90	10-13	Char-Codes for Room Fountains
00002E94	14-17	Char-Codes for Room Living Statues
00002E98	18-1B	Char-Codes for Hand Weapon
00002E9C	1C-1F	Char-Codes for Ranged Weapon
00002EA0	20-23	Char-Codes for Shield
00002EA4	24-27	Char-Codes for Armor
00002EA8	28-2B	Char-Codes for Chest
00002EAC	2C-2F	Char-Codes for Gold
00002EB0	30-33	Char-Codes for Map
00002EB4	34-37	Char-Codes for Dead Thing

Miscellaneous Global Values

00002EB8	38	(01)	Initial Value (x 10) for Experience assigned to Monsters in Monster List
00002EB9	39	(03)	Progression (x 10) of 1) Monster Experience Points, 2) Gold amounts in Rooms and Vaults
00002EBA	3A	(00)	
00002EBB	3B-3C	(01A9)	Party location last time game was saved. Same value as Sector 002E, bytes F8-F9 for hallways. If a room is entered it represents the last hallway location.

Game Option Soft Keys

Game	Sector 0050			
Byte #	Bytes			
00002EBD	3D	(42)	B	Break Door
00002EBE	3E	(43)	C	Check for hidden Doors
00002EBF	3F	(46)	F	Fire ranged weapon
00002EC0	40	(57)	W	Weapon change
00002EC1	41	(4C)	L	Listen for Monsters
00002EC2	42	(4D)	M	Map check
00002EC3	43	(4F)	O	Change Party Order
00002EC4	44	(4B)	K	Keep Game i.e. Save Game
00002EC5	45	(54)	T	Trade Items
00002EC6	46	(55)	U	Use Item
00002EC7	47	(4E)	N	Negotiate with Monsters

Note: These Command soft-keys can be changed, but this will not update the TOD Module Help Screen. Also, care needs to be taken to prevent unexpected results. E.g. changing Keep to Save (game) will invoke the Save Game Menu every time you press 'S' to go South!

Miscellaneous Global Values

Game	Sector 0050		
Byte #	Bytes		
00002EC8	48-4C	(00)	(Any data written here immediately reverts back to "00")
00002ECD	4D	(02)	AV for "Hands". This value is written to Sector 002E (Player Stats) as Players' AV if no weapon is present, but reset by module to "2" on use.
00002ECE	4E	(14)	Maximum Gold in Vaults (First Floor)
00002ECF	4F	(1E)	? Uncertain

Game Key Words

Game	Sector 0050			
Byte #	Bytes	Word	Max. Length	Char-Code
00002ED0	**50-5F**	FOUNTAIN	18 bytes	
00002EE0	**60-6F**	LIVING STATUE	16 bytes	
00002EF0	**70-7F**	*GOLD PIECES	16 bytes	
00002F00	**80-8F**	*MAGICAL ITEMS	16 bytes	
00002F10	**90-9B**	*PLAYER	12 bytes	30
00002F1C	**9C-A7**	*PARTY	12 bytes	31
00002F28	**A8-B3**	*EXPERIENCE	12 bytes	32
00002F34	**B4-BF**	*LEVEL	12 bytes	33
00002F40	**C0-CB**	*RANGED	12 bytes	34
00002F4C	**CC-D7**	*WEAPON	12 bytes	35
00002F58	**D8-E3**	*ARMOR	12 bytes	36
00002F64	**E4-EF**	*SHIELD	12 bytes	37
00002F70	**F0-FB**	*MONSTER	12 bytes	38
00002F7C	**FC-07**	TRAP	12 bytes	39 & 40
	Sector 0051 - *Game Key Words*			
00002F88	**08-13**	*RATION	12 bytes	41 & 3A
00002F94	**14-1F**	*HEALING	12 bytes	42 & 3B
00002FA0	**20-2B**	*SPEED	12 bytes	43 & 3C
00002FAC	**2C-37**	RESISTANCE	12 bytes	44 & 3D
00002FB8	**38-43**	MAGIC	12 bytes	45 & 3E
00002FC4	**44-4F**	ALL	12 bytes	46 & 3F
00002FD0	**50-5B**	PROTECTION	12 bytes	47
00002FDC	**5C-67**	MOBILITY	12 bytes	48
00002FE8	**68-73**	SPECIAL	12 bytes	49
00002FF4	**74-7F**	CLASS	12 bytes	4A
00003000	**80-8B**	ATTACK	12 bytes	4B
0000300C	**8C-97**	DAMAGE	12 bytes	4C
00003018	**98-A3**	COMBAT	12 bytes	4D
00003024	**A4-AF**	ROOM	12 bytes	4E
00003030	**B0-BB**	AVAILABILITY	12 bytes	4F
0000303C	**BC-C7**	INFORMATION	12 bytes	50
00003048	**C8-D3**	POWER CHANCE	12 bytes	51
00003054	**D4-DF**	PROBABILITY	12 bytes	52
00003060	**E0-EB**	BRIBABILITY	12 bytes	53

Game Key Words - continued

Game	Sector 0051			
Byte #	Bytes	Word	Max. Length	Char-Code
0000306C	**EC-F7**	GOLD	12 bytes	54
00003078	**F8-03**	HIT POINTS	12 bytes	55
	Sector 0052 - *Game Key Words*			
00003084	**04-0F**	WANDERING	12 bytes	56
00003090	**10-1B**	LUCK	12 bytes	57
0000309C	**1C-27**	CONSUMPTION	12 bytes	58
000030A8	**28-33**	BONUS	12 bytes	59
000030B4	**34-3F**	INTERVAL	12 bytes	5A

These Keywords are used to describe Spells & Effects. Keywords preceded by an '*' also change the corresponding keyword in the Status Reports, Help Screen & General Store; so care should be exercised. Changing others, e.g. Mobility, can lead to inconsistency in term use between Spells and Reports.

Some Report Screens are formatted for specific word lengths (E.g. EXPERIENCE). There is a formatting relationship with the keywords "EXPERIENCE, LEVEL and WEAPON" and another with "ARMOR, SHIELD and PROTECTION". Best to keep them at their current length or add preceding space characters (in some instances). Check all screens post changes.

List of the World's Known Spells Dialogue - Recognized by the TOD Module

Game	Sector 0052			
Byte #	**Bytes**	**Code**	**Spells**	**Effect**
000030C0	**40-42**	305520	00 & 01	PLAYER HIT POINTS
000030C3	**43-45**	304C20	04 & 05	PLAYER DAMAGE
000030C6	**46-48**	303647	08 & 09	PLAYER ARMOR PROTECTION
000030C9	**49-4B**	30354C	0C & 0D	PLAYER WEAPON DAMAGE
000030CC	**4C-4E**	303659	10 & 11	PLAYER ARMOR BONUS
000030CF	**4F-51**	303559	14 & 15	PLAYER WEAPON BONUS
000030D2	**52-54**	305720	18 & 19	PLAYER LUCK
000030D5	**55-57**	303220	1C & 1D	PLAYER EXPERIENCE
000030D8	**58-5A**	303320	20 & 21	PLAYER LEVEL
000030DB	**5B-5D**	2B2B2B	(not used)	(Filled with 2B='+' plus sign symbol)
000030DE	**5E-60**	46304C	28 & 29	ALL PLAYER DAMAGE
000030E1	**61-63**	315420	2C & 2D	PARTY GOLD
000030E4	**64-66**	314120	30 & 31	PARTY RATION
000030E7	**67-69**	31354C	34 & 35	PARTY WEAPON AVAILABILITY
000030EA	**6A-6C**	2B2B2B	(not used)	
000030ED	**6D-6F**	314D43	3C & 3D	PARTY COMBAT SPEED
000030F0	**70-72**	563852	40 & 41	WANDERING MONSTER PROBABILITY
000030F3	**73-75**	41585A	44 & 45	RATION CONSUMPTION INTERVAL
000030F6	**76-78**	31425A	48 & 49	PARTY HEALING INTERVAL
000030F9	**79-7B**	383647	4C & 4D	MONSTER ARMOR PROTECTION
000030FC	**7C-7E**	384B4A	50 & 51	MONSTER ATTACK CLASS
000030FF	**7F-81**	384B4C	54 & 55	MONSTER ATTACK DAMAGE
00003102	**82-84**	384951	58 & 59	MONSTER SPECIAL POWER CHANCE
00003105	**85-87**	385320	5C & 5D	MONSTER BRIBABILITY
00003108	**88-8A**	384820	60 & 61	MONSTER MOBILITY
0000310B	**8B-8D**	384544	64 & 65	MONSTER MAGIC RESISTANCE
0000310E	**8E-90**	384D43	68 & 69	MONSTER COMBAT SPEED
00003111	**91-93**	463855	6C & 6D	ALL MONSTER HIT POINTS
00003114	**94-96**	385520	70 & 71	MONSTER HIT POINTS
00003117	**97-99**	4E3850	74 & 75	ROOM MONSTER INFORMATION (not used by game)
0000311A	**9A-9C**	2B2B2B	(not used)	
0000311D	**9D-9F**	2B2B2B	(not used)	

Note: Dialog for Spells above 71 is generated by the TOD Module.

00003120	A0-AB	'CHEST'	12 Bytes
0000312C	AC-B7	'VAULT'	12 Bytes

Map of Current Floor (including a Saved Game)

Game	Sector 0052		
Byte #	**Bytes**	**Map Row**	**Border**
00003138	**B8-D2**	1st Row of Map	**D3-D6**
00003157	**D7-F2**	2nd Row of Map	**F3-F6**
00003177	**F7-12**	3rd Row of Map	**13-16**
	Sector 0053		
00003197	**17-32**	4th Row of Map	**33-36**
000031B7	**37-52**	5th Row of Map	**53-56**
000031D7	**57-72**	6th Row of Map	**73-76**
000031F7	**77-92**	7th Row of Map	**93-96**
00003217	**97-B2**	8th Row of Map	**B3-B6**
00003237	**B7-D2**	9th Row of Map	**D3-D6**
00003257	**D7-F2**	10th Row of Map	**F3-F6**
00003277	**F7-12**	11th Row of Map	**13-16**
	Sector 0054		
00003297	**17-32**	12th Row of Map	**33-36**
000032B7	**37-52**	13th Row of Map	**53-56**
000032D7	**57-72**	14th Row of Map	**73-76**
000032F7	**77-92**	15th Row of Map	**93-96**
00003317	**97-B2**	16th Row of Map	**B3-B6**
00003337	**B7-D1**	17th Row of Map	

Appendix IV - TOD Database Reference

Note: The Map always represents the current Floor the Party is located. The Saved Map totals 17 Rows and reads from Top to Bottom, Left to Right. The first and last rows of a Saved Map are 27 bytes or characters wide, the other 15 Rows are 28 bytes. The Space Character '20' is used to provide a Border 2 bytes or characters wide, to the left and right of each row (except before the first & end of the last row). Thus each Row totals 32 Characters.

This Map is always present upon entering a Floor, part of what is going on in the background when you Descend or Ascend a Floor, but is generally invisible with both Foreground & Background Colors being set to Gray. (Codes 70 - 7F) As a Floor is explored the 60 series Codes are replaced with 60 series Codes and thus become 'visible'. When a Map is found the module changes the Map to Blue on Gray.

When you are in the Store on the Ground Floor, the Map of the first Floor is present.

Saved Game File Structure - VDP Block (files)

Game	Sector 0054			
Byte #	Bytes			
00003352	**D2**	0	I/O Op-Code	06 = Save Option
00003353	**D3**	1	Flag/ Status	0A = 1010 Binary (1=Variable length, 0=Display, 1=Output, 0=Sequential)
00003354	**D4-D5**	2&3	Data Buffer Address	0400
00003356	**D6**	4	Logical Record Length	80 = Display
00003357	**D7**	5	Character Count	80 Bytes
00003358	**D8-D9**	6&7	Record #	33 = 51 Sectors
0000335A	**DA**	8	Screen Offset	00
0000335B	**DB**	9	File Name Length	0A = 10 Bytes "DSK1.QUEST" = 10 Bytes (up to 1C or 28 Characters)
0000335C	**DC-F7**	10	File Description	Device/ Filename (up to 28 Characters in length)
00003378	**F8-FF**		Character Locations	(at Start of Game = Ground Floor)

<u>General Database Breakdown</u>	Sectors:		Total
Game Title & Description:	24-25		2
Graphics: (21 Sectors)	22-23	(N,S,E,W symbols; ranged & magical attack graphics)	2
	26	(Characters from saved game)	1
	27-28	ASCII Character definitions	2
	29	Map Graphics	1
	2A-2D	Hallway Graphic space)	4
	2F-30	(Party A/ D) 1 Sector),	1
	37-3A	(Monster A/ D)	4
	4C-50	Rooms & Room Contents graphics	4
	53-54	Current Map	2
Lists & Game Settings:			17
"Saved" Game Information			<u>11</u>
Total Game Information (Current & Saved)			51 Sectors

Total Game Bytes = Hex 32FF = 13056 Bytes