

A MANUAL FOR EAMON ADVENTURE DESIGNERS

By Donald Brown

EAMON is the computerized fantasy role-playing game developed by Donald Brown. This manual has been written for those stalwart people who are tired of having their characters killed in the many adventures written for the system, and want revenge by creating their own death traps. It is assumed that you are already familiar with the gaming system and the information included in the Player's manual.

Most of the adventures that are written for the Eamon system have been similar to the "Adventure" game that was created by Don Woods and Willie Crowther of MIT, such as the Beginner's Cave included on the master diskette. This manual will both help you design your own scenarios in that type of game but will also permit you to meld any other type of game with the system that you wish.

HOOKING UP WITH EAMON or SENDING ADVENTURERS TO THEIR DEATH FOR FUN AND PROFIT

Under the Eamon rules, it takes only a few numbers to completely describe a character. Between adventures, all of the player characters, (or adventurers) that the Master system knows about are stored in a file on the Master that is called "CHARACTERS". It is a random-access file with a length of 150. Record 0 holds the number of records used in the file, and all subsequent records may hold one character. If the first string in the record is null ("") then the character in that record has been deleted and the record may be reused to store a new character. The data held in each record is:

A string that has the name. The next numbers stored are (in order) player Hardiness, Agility, Charisma, the four spell abilities (Blast, Heal, Speed, and Power), the five weapon abilities (Axe, Bow, Club, Spear, and Sword), the player's Armour Expertise, the player's sex as a string ("M" or "F"), gold pieces carried, gold pieces in the bank, the player's armour class (Leather=2, Chain=4, Plate=6, with one added for a shield), then for each of a player's four weapons the name of the weapon, the weapon type, the weapon complexity, the weapon dice, and the weapon's sides per die. If a player does not have four weapons, his weapons will be first, and all other weapon names will be "NONE". Except for name, sex, and weapon names all of the information is stored as an integer number, with all probability numbers stored as percentages (perfect=100).

When a player leaves the Main Hall to go on an Adventure, first his character is deleted from the CHARACTERS file, and then the player is prompted to change diskettes. The program then tries to read a program name from the file EAMON.NAME on the adventure disk.

If this name is found, a file called FRESH MEAT is opened on the disk and the data that was in the record of the CHARACTERS file is written into that file, preceeded by the record number that the character previously resided in. Finally, the program given in the EAMON.NAME file is run.

Once the adventure is over, control must be returned to the main Eamon programs. If the adventure ended in the character's death, all that must be done is deleting a file called "THE ADVENTURER" from the master diskette and running MAIN HALL. (You may instead directly run THE WONDERFUL WORLD OF EAMON and then do not need to delete the file. If you do want to go the MAIN HALL route, it might be advisable to open THE ADVENTURER before deleting it, to make sure it will really be there.)

If the character survives your adventurer, you will have to recreate him into the CHARACTERS file. The Main Program logic is already writing the character his own character record now, but you may want to change this if you have a special reason for it. Normally this is simply writing in the new information of the character into the old record given in FRESH MEAT. However if your program has the ability to quit for a while and come back later, it is possible for a new character to be stored in the old record. In this case you should search the CHARACTERS file for a free record to store the character, and write it in there (for an example of how this is done, list the NEW CHARACTERS program on your master diskette).

Once you have stored the character into the CHARACTERS file, you must re-create a file called THE ADVENTURER which has two pieces of data in it--the name of the character, and the record he resides in of the CHARACTERS file. This too, is already done by the logic of the Main program, but it may be necessary for you to know what is being done, in case you want to change it. Once this is done, run MAIN HALL to finish. (Also currently done.)

USE OF THE DUNGEON DESIGN DISKETTE

The Dungeon Design Diskette is designed to make the job of entering new Adventure-like scenarios much easier. It includes a base to work from for your program, as well as a simple means of entering the data into the text files needed by that base program.

Version 5.0 of the Dungeon Designer Diskette will contain two versions of the Main Adventure program. One is called MAIN PGM (or BASE PROGRAM 2.0) and the other is MAIN PGM 10 (or BASE PROGRAM 10).

Version 6.0 will only contain MAIN PGM, but will contain an extra file called CONVERT MAIN>MAIN 10. This is a text file that can be executed (using the DOS EXEC command) to convert MAIN PGM to a ten direction version. To do this simply load in MAIN PGM and type EXEC CONVERT MAIN>MAIN 10. You will see some Applesoft prompts. When it stops, you will have a ten direction version of

the MAIN PGM. You may then save this on your adventure diskette.

The first step in creating your adventure is initializing the diskette. The program on the DDD (Dungeon Design Diskette) will do several things for you---it will INIT the diskette, put your program's name into EAMON.NAME, and put in the starting data needed for the dungeon editing program to work. It also creates a boot program that identifies this diskette as an Eamon adventure by you! You will be asked for an adventure number, this must be an integer from 1 to 254, since it is used as the disk volume. It is not important what number you assign, it will be assigned to the next valid number - once you have submitted it to the library.

When you have initialized your diskette, you should delete the text file SAVE LEADIN PROGRAM HERE. Then re-insert the DDD and load LEADIN PROGRAM. The diskettes should be switched again and you will type in SAVE (your adventure name). You must spell the name to be saved -in () above exactly as you did when you typed in the name of the adventure during the initializing process. Then re-insert the DDD and type LOAD MAIN PGM or LOAD MAIN PGM 10 (depending on whether you answered 6 or 10 directions during the initialize process. The MAIN PGM 10 should be used if you answered 10 to the number of directions. Otherwise, use MAIN PGM. Now re-insert your new adventure diskette and type SAVE MAIN PGM. Note --- here you use 'MAIN PGM' regardless of which program you loaded. You are now ready to begin entering data for the adventure. Insert the DDD and run the program 'DUNGEON EDIT'. This program is the one that puts your basic dungeon design into the files.

A few general remarks are in order now--First of all, you should always have your dungeon designed before entering it. Decide what rooms you have, how they connect, what monsters inhabit those rooms (and what their attributes are), and what treasures are sprinkled about.

For each of the four things you can enter (ROOM, ARTIFACT, EFFECT, and MONSTER), you can either add a new one onto the end of the list, or edit one already there. You cannot delete a thing already there (though you can replace it through editing). You also must not go beyond 100 of any of the things (though the total can go over 100). Actually you will run out of disk space long before you hit the limit of 100 on these items. The more practical limits of the system are: (-in general-)
Rooms: 50-85 Artifacts: 35-50;
Effects: 0-25; Monsters: 12-35.
If you do go over the 100 limit on anything you will not be warned of your error, but you will not get the results you wanted.

Adding and Editing will be almost identical for all four data types, except that in 'editing' your old entry will be placed after the cursor at the start of entries. Every time you are to do more entry than just hitting one

key, the entry will be done with a special input routine. It appears to be the standard Apple input on first glance, but it has many significant differences. First of all, the old Escape-key editing features do not work. Instead, the following control keys do things--

ESC--this returns, accepting both what is before and after the flashing cursor. It is very useful while editing since if you simply hit ESC no change will be made.

RETURN--this also does a return, except only accepting what is before the cursor.

CTRL-B--this moves the cursor to the first character of your input.

CTRL-E--this moves the cursor to the last character of your input.

CTRL-D--this deletes the character that the cursor is sitting on, bringing the characters after the cursor forward one space.

CTRL-I--this inserts a space where the cursor is, moving all characters after the cursor back one space. This is good if you are entering a line that you want to look good as it wraps around your 40-column screen.

<-,> (FORWARD & BACKWARD ARROWS)-- these move the cursor back or forward by one character.

Your entries cannot go over 250 characters, and you should not use quotation marks ("). You may use commas and colons in descriptions only. Do not have any trailing spaces on the names of items. (Applesoft regards "LION" as being different from "LION ", though your player may not recognize the difference. Additionally, if you are entering a number, enter only a number, do not include any leading or trailing spaces or other extraneous characters. The program won't like them and will simply spit them back at you.

Another thing to watch out for is names on artifacts that begin with a number. The program will accept them and you can use them, but when you try to give the item to a monster the program will not understand. For example assume you want to put an 8 inch knife in your adventure. When adventurer picks it up and tries to give it to his friend the Doctor, the Doctor will think he is being given 8 gold pieces. That is because the program looks at the artifact name and because it begins with a number, it mis-understands.

For every room, you will need to give eight pieces of data. First is the room name, which cannot go over 39 characters. In the program it will be printed as: "YOU ARE (room name)", so you should use names like, "AT THE CAVE ENTRANCE". Secondly you will need to give a room description. It is

not preceded by anything, so it must be a full and complete description. If your description is longer than 40 characters, you must pad it with spaces so that when the description wraps around the Apple's 40-column screen, the breaks are between words. Finally, you will have to give the room numbers that you can get to from that room in each direction. A special code has been developed--if you give a room of 0, you can never move that direction. If you give positive direction, there's an open connection. Negative numbers are special and usually indicate a secret passage. Another special code is -99 which indicates the exit to home. Negative numbers can be easily made to have special results by altering the base program (see below).

Artifacts are somewhat similar. An artifact is any non-living thing that is in the dungeon. In addition to what you might normally think of as the artifacts you want to include: (gold, silver, statues), you must also have as an artifact all weapons used by your monsters, as well as a dead body for every monster.

For each artifact, you will again need a name (this time just the normal name, such as "GOLD COINS") and a full description. You will also need to give the item's room that it starts in, its value in gold pieces, its type and its weight. The room is usually a positive number, however if the item isn't in the dungeon yet (such as a dead body) you should assign a room of zero, and if the item starts by being carried by the player, its room is -1.

There are four types of artifacts. Type 0 is a treasure with a value that will not change with the player's charisma, such as a pile of gold coins. Type 1 is a treasure whose value will vary with the player's charisma, such as a Persian rug. Types 2 and 3 are weapons, with 2 being a weapon that can be bought at the Main Hall, and 3 a special weapon.

If your artifact is a weapon (type 2 or 3), you will also have to enter some more information on it. First is the weapon complexity, then is the weapon type (1=axe, 2=bow, 3=mace, 4=spear, 5=sword), then the weapon's damage with first the number of 'dice' thrown for that damage, then the sides per die.

The other major type of data to be put in your files is monsters, which are any living (or animate) things in the dungeon. Monsters are similar to characters, however they are assumed to have their full armour expertise and know all weapons equally well. For each monster you will need the name, the description, then hardiness (as with a player), agility, friendliness (the percentage chance of making friends with a character of a charisma of 10), courage (will flee from a fight after he has received that percentage of his hardiness in hits, on the average), room starts in (may be zero if in a chest or other special thing activates him), his body weight, his special defensive odds (normally 0, but magic

or size/speed may make some monsters harder to hit), defensive odds are a %. Armour (hits absorbed or stopped - per blow. This may be things such as a furry skin or magical effects), and the weapon number (a pointer to some artifact. If the weapon number is 0 it is assumed natural weapons; claws or teeth. If the weapon number is -1 then the monster isn't carrying a weapon). You will then have to give for that weapon the monster's complete chance to strike a blow with it, and its sides and dice of damage. These numbers do not have to be consistent with the numbers given in the artifact list; a monster may know how to use his weapon better than someone who doesn't know some secret, and the chance of hitting is of course affected by the monster's weapon expertise and other information not stored.

There is a fourth thing that can be entered called an "EFFECT". It is there for your own special use--it permits you to store some strings on disk so that it can be called in quickly. The code to read effects is not in place in the base program (Main Pgm). To read an effect (number E, for example) use a routine such as:
PRINT DK\$;"READ EAMON.DESC,R";E+200
: INPUT A\$: PRINT A\$: PRINT DK\$.
NOTE: This routine also prints the effect (A\$) that it reads. For further examples of its use you might want to check out The Beginners Cave. It is also included as room for expansion of the DDD.

Once you have put all of your data into your files, you will probably want to see what you entered, to catch errors and get a good overview. There is a program included on the DDD that will list all of your data in a simple, organized fashion. It will list all of your rooms, artifacts, and monsters in order, also pointing out what sort of "links" have been set up (what is the room name you are moving into, what is the name of the artifact that the monster uses as a weapon, etc.) If one of these links goes to the wrong thing, there's an error! If you want to put this output to a printer, you must modify lines 9000-9999 of the program DUNGEON LIST to start your printer and set the value of PL to the line length of your printer, less one (39 for the Apple's screen).

There is also a program included called DUNGEON LIST (OLD) which was written to work with the older system of files where EAMON.MONSTERS and EAMON.ARTIFACTS were sequential files. Although it might be very useful to examine other people's dungeons after you have played in them to learn how to design your own, it goes without saying that only a cur and a scoundrel would list someone's dungeon before playing to avoid dangers!

ADAPTING THE BASE DUNGEON PROGRAM

All of the work above was to put your dungeon into a format that the computer can use. It couples with a program called the BASE DUNGEON

PROGRAM (or on later versions called MAIN PGM or MAIN PGM 10). If no sliding doors, things hidden inside of other things, etc. are needed, and you are using the original DDD, you can simply save the base program on your diskette and have a program that prints the explanation or mission run this program. If you have a newer version of the DDD and have already saved the MAIN PGM and your (name of adventure) program, you are all ready to go.

However, if you want special effects such as a sword that teleports the user to another room at random times (gee, I may use that) these explanations should help you.

Lines 100-999 are the main loop. Every time a command is gotten, those lines are run through. If you want to have something done (or checked for) every turn, it should be put in lines 500-900.

Lines 1000-1999 reads in monsters and artifacts from disk, as well as doing other initializing. If you want to add a new command, you will have to change line 1910 (increase the number in the DATA statement), 1920 (add the new verbs--no spaces are permitted), and line 290 (add the line numbers to go to). If you want special things to happen at the start, such as a fee from the player's gold for some item, do it in lines 1150-1890.

Lines 2000-2999 are the closing routines. When this is entered, if the variable DIE has a non-zero value the player didn't survive. Lines 2100 thru 2290 are for your additions.

Lines 3000-3999 are the movement commands. If you want to magically move the player, set R2 to the number of the room to enter and go to 3500. If something happens so as to make the monsters reconsider their reactions to the player, a GOSUB 3600 will check the reactions of all monsters who are unaligned (see below). If you have special results and/or conditions for movement and have thus given negative room numbers in the editor, the place to check is in lines 3050-3490.

Lines 4000-4999 are to get things. Lines 4200-4899 are the place to add special results when getting some artifact. You may place synonyms in lines 4030-4110 (such as, 4030 IF S#="BAT" THEN S#="VAMPIRE BAT"). Finally, if a command MUST have a subject, GOSUB 4900 will ensure one is gotten.

Lines 7000-7999 are the attack commands and subroutines. Lines 7700-7999 are to kill monster M. If things happen when a monster dies (such as dropping something from the body), it should be put in this section.

Lines 13000-13999 are the Power Spell. As told in the Players manual, this spell can do anything at all--feel free to through out this section and add your own.

Lines 16000-16999 are the say command, which is very useful for 'words of power' or some other strange effects.

Most of the variables are self-explanatory and also can be changed locally if desired, but a description of some of the others are--

AC - Armour class of player

AD%(*,*) - Artifact data

The first subscript is the number of the artifact, and the key for the second is:

- 1=Value
- 2=Type
- 3=Weight
- 4=Room
- 5=Complexity
- 6=Weapon Type
- 7=Dice
- 8=Sides
- 9=Flag if seen

AE - Armour expertise

AN%(*) - Name of artifacts

BANK - Gold player has in bank

C - Holds number of command given

C%(*) - Verbs program responds to

CH - Player charisma

CZ\$ - last command given

DF - defender

DIE - Logical flag, 1=player died

DK\$ - holds CTRL-D for disk commands

DR%(*) - room moved in each direction

EA - effect of armour on odds-to-hit

FD%(*) - full damage of side in combat

FR - fumbler roll/friend rating

GOLD - gold player has on person

HIT - logical flag if hit in combat

INC - logical flag if ability increased

LK - logical flag if 'looked' already

MD%(*,*) - monster data

First subscript is monster number,

second key is:

- 1=Hardiness
- 2=Agility
- 3=Friendliness
- 4=Courage
- 5=Room
- 6=Weight
- 7=Defensive odds (%)
- 8=Armour
- 9=Weapon #
- 10=Odds to hit (%)
- 11=W Dice
- 12=W Sides
- 13=Damage
- 14=Reaction
 - 0-not met
 - 1-unfriendly
 - 2-neutral
 - 3-friendly

MN%(*) - Name of monster

MR - Monster morale

NA - Number of artifacts

NBTL - logical flag if in battle

NC - Number of commands

NM - Number of monsters

NW - Total count of weapons in game

NZ - Number artifacts not player weapon

OF - Number of offensive monster

RAISE - logical flag if power raised

REC - Player record in char file

RL - Random number 1-100

ROOM - Room player currently in

RR - Random number 1-100 for power

S\$ - Subject of command given

S2%(*) - Current spell ability

SA%(*) - Total spell ability

SEX\$ - Holds 'M' or 'F' for player

SPD - Number of turns speed spell to go

SUC - logical flag if spell succeeded

TD%(*) - Damage taken for side

TP - Total price of treasure

U\$ - Verb of command

UX%(*) - Flags if player been in room

WA%(*) - Player's weapon ability

WD%(*) - For weapon, dice of damage
WN\$(*) - Name of player's weapon
WO%(*) - Weapon complexity
WP%(*) - Weapon pointer (in close)
WS%(*) - Sides/die of damage for weapon
WT - Weight player carrying
WT%(*) - Weapon type
WZ - Number of weapons player brought

Once you have "played-out" other people's adventures, it would definitely be a good idea to tear it apart to see how other people have adapted this program to their own use. Also, if I might give a few suggestions to new Eamon dungeon designers-

1. Be fair to the adventurer. Don't try to stack the odds totally against him. A good rule of thumb is that an adventurer can lick about five times his own Hardiness in opponents, with allies Subtracting their Hardiness from the opposition. If you continually design pure death traps (and don't reward the successful outrageously), people aren't going to want to send their carefully-built up characters through your dungeons. On the other hand, if your dungeons are just big give-aways, they will quickly become boring.

2. You can place one or two traps of the zap-you're dead type (such as the book in the Beginner's Cave) so long as they are not overdone and are not required to survive them to get out. Thus, if your only way out of the tunnels is drinking a potion that half of the time teleports you away, the other half poisons you, it isn't fair.

3. Particularly to those who have designed dungeons for non-computerized role-playing games, remember that there is only one adventurer going in, not an army! No matter how great a character is, he cannot by himself handle a dozen thugs.

4. Last but not least, don't be afraid to break any of these rules. If you truly believe that your dungeon will be better, do anything you please. The worst that will happen is that people will not play in it and you'll have to change a few things.

.FF

The process of creating your own adventure in Eamon will require that you have a basic knowledge of programming in Applesoft and in operating your system as far as being able to load and save programs as well as copying files and diskettes. If you are unfamiliar with these operations, you should review them until you are comfortable doing these functions.

The following is a summary of the steps to follow to create an adventure of your own. This summary is provided to allow you an easier step-by-step method of creating your scenario.

1. Be sure you have everything you need. The things you will need are:
 - a. A theme for the adventure. This is usually the hardest part.

- b. A map of the setting for the adventure
- c. A list of all the monsters that will appear.
- d. A list of all the artifacts you will need.
- e. The DDD (Dungeon Designer's Diskette)

2. The next step is to initialize your adventure diskette. This is done by booting on the DDD and selecting the initialize function. You will be requested to replace the DDD with a blank diskette. Then you will be asked for the name of the adventure, the adventure number and the author's name.

THE DISKETTE THAT IS SITTING IN THE DISK DRIVE AFTER THESE QUESTIONS ARE ANSWERED WILL BE INITIALIZED. FOR THIS REASON, YOU SHOULD ALWAYS KEEP YOUR DDD WRITE PROTECTED.

3. When the diskette has been initialized, you'll need to copy two programs from the DDD to your adventure diskette. The first one that should be copied is LEADIN PROGRAM. Copy this one by:

- a. Insert the DDD and type in:
LOAD LEADIN PROGRAM
- b. Insert the adventure & type
DELETE SAVE LEADIN PROGRAM HERE
- c. Now type: SAVE (whatever you called your adventure)

The name that you use in the save command must be the exact name that you used when you initialized the diskette.

The second program that you have to copy from the DDD to the adventure diskette is the main adventure program. The name that this program goes by on the DDD may be one of the following:

```
BASE DUNGEON PROGRAM
BASE PROGRAM 2.0
MAIN PGM
```

This will depend on which version of the DDD that you have. If you answered the question "six or ten directions" as 10, then you will have to use the ten direction version of the main program.

Some versions of the DDD do not have this capability. However, if you did get a question on how many directions you wanted, your designer does have this capability. If you are using version 6.0, and you want to use ten directions, you have to create a ten direction version of the main program. This is a very simple process. Simply load the MAIN PGM program into memory, and type : EXEC CONVERT MAIN>MAIN 10.

When the process is complete, you will have a ten direction version in memory. You may then save this on the adventure diskette. When you have the appropriate program in memory, insert the adventure diskette and save it under the name MAIN PGM.

4. The next step is to add the data or information about your adventure. All

items are added using the DUNGEON EDIT program. Boot on the DDD and select modify an adventure. The edit program is run and then requests you to insert your adventure diskette. You will then add all of your rooms, artifacts, effects and monsters. This may take several days, so whenever you get tired of entering, select quit option. All of your items will be saved and you can pick up where you left off.

5. After all your rooms, artifacts, etc have been entered, you may make your program changes to MAIN PGM. These will be all of the things that you want to be special about your adventure.

6. The next phase will be to test your adventure. The best way to do this is to be sure the PRINT DK#"DELETE FRESH MEAT" statement has been deleted from the MAIN PGM, (.try line 1055 or 1050) and that your main program has been saved to the adventure diskette after making your changes. Then boot the Eamon Master diskette. Select the character to be used to test with and take him into the Main Hall and select adventure. Insert adventure diskette and you should now begin your test. From that time on, to repeat a test you should be able to simply run MAIN PGM. (Or if MAIN PGM is already in memory, simply type RUN)

Another helpful hint: to restart an adventure if it bombs off you should be able to type GOTO 210.

7. When you are sure your adventure is ready you could have a friend test it. When the adventure is completely ready, send a copy of it to:

Tom Zuchowski
The Eamon Adventurer's Guild
7625 Hawkhaven Dr.
Clemmons, NC 27012
(919)766-7490

It will then be an "official" Eamon Adventure and may be distributed.

If you have any questions or problems designing an adventure, write to the above address. We will do our best to help you get back on the right track.

THE END