>>>>

How to *HACK* into games

By Nathan Anderson

Includes Pascal Source Code !

Contents

Introduction

So you want to hack game files eh?

I suppose many readers may be wondering why they should bother?
Successfully hacking into a game can be interesting and even fun for some people.
You may discover something that was disabled in the game like different weapons, graphics or even levels.
Take for example the game Doom 2, according to some, there is actually more than 2 secret levels (which you can't normally/easily access when you play the game!).
(Hat tipped to Richard for that one)

Generally there are several reasons that people decide to create a hack for a game,

- To gain advantages in a difficult game, also known as cheating....
- To disadvantage yourself in an easy game, also known as creating a handicap.
- Just for the hell of it. In a game that you have beaten, a good hack can give it new life.
- To gain 15 minutes of fame - if your hack gets published in a computer magazine.
- To gain money from the computer magazine who printed your hack.

What equipment is needed?
The obvious equipment would be:
A computer and one or more games.

You will also need:

- A hex / disk editor, (and know how to use it)
- Some knowledge of Hexadecimal and Machine code, (this manual is a start)
- A pen and paper,. (to make notes of your discoveries)
- A lot of time! (don't always expect the cheat to work first time)

Notes: This manual will simply show how to do the most basic of hack's, mainly of save game files.
Since most games vary in there structure - creating more advanced hacks would be impossible to describe (and may be beyond my knowledge).
If you wish attempt more advanced game hacks I suggest learning how to program in machine code and assembler, as more complex hack's require you to physically rewrite parts of a game whilst leaving other parts intact.

Explanation of Hexadecimal
In order to change a save game file, you need to know how the numbers are stored in the files.

Hexadecimal, which is abbreviated as hex when used in computer terms, is a different counting base.
Usually us humans use base -10. This is that from 1 to 10 there is 10 numbers eg.
1 2 3 4 5 6 7 8 9 10

But hexadecimal is base - 16. To count to 10 you go through 16 numbers eg.
1 2 3 4 5 6 7 8 9 a b c d e f 10

As computers use Binary, which is base - 2, eg.
1 10
this is very messy to look at, it would be converted to hexadecimal to make it easier to program in machine code.

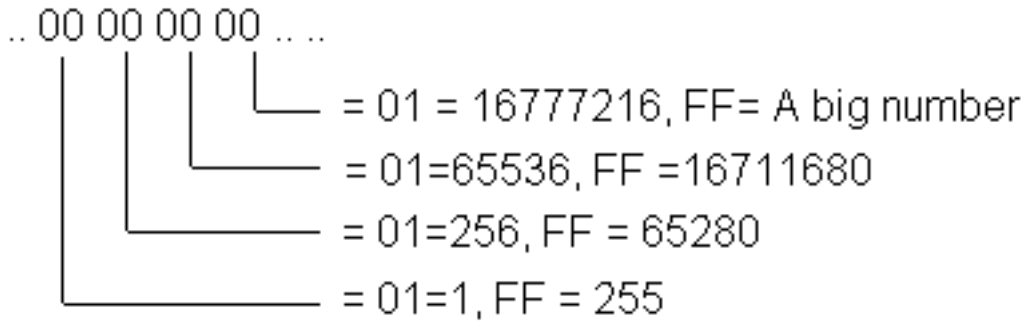
DOS only uses one byte per character, one byte equalling 8 binary spaces or 0000 0000. So the highest number you can show in binary is 1111 1111 which equals FF in hexadecimal.
This is only effective up to the number 255 which is what the above binary and hexadecimal numbers equal.

Games store numbers in hexadecimal format instead of character format the idea being that a number like 255 stored as hexadecimal uses less disk space than as characters.
For example, 255 stored as hexadecimal would be:
FF
which equals a single character

But if you stored 255 as characters it would take up 3 bytes:
.. 32 35 35 .. .. 2 5 5
(hex characters)

Note: If you are already computer competent, then you may wonder why I am referring to FF as meaning 255, while when RAM is concerned 256 is used instead; this is because mathematically FF does equal 255, but in computer terms, if you include 00 as equalling one unit, then FF would make a total of 256 separate units.

You can't have more than double digits with a hex number, so what if you need a number higher than 255? (or FF in hex)
Well there is a tricky formula used to store higher numbers in hex, here is a rough diagram showing it:



The following are some examples, how they were worked out is in brackets.
So to write the number 1 as hex you would format it as:
01 00 00 00 (1 x 1)
To write the number 255 as hex you would format it as:
FF 00 00 00 (255 x 1)
To write the number 256 as hex you would format it as:
00 01 00 00 (1 x 256)
To write the number 257 as hex you would format it as:
01 01 00 00 (1 x1, 1 x 256)
To write the number 65280 as hex you would format it as:
00 FF 00 00 (0x1, 255x256)
To write the number 65536 as hex you would format it as:
00 00 01 00 (0x1, 0x256, 1x65536)
To write the number 16711680 as hex you would format it as:
00 00 FF 00 (0x1, 0x256, 255x65536)
To write the number 16777216 as hex you would format it as:
00 00 00 01 (0x1, 0x256, 0x65536, 1x16777216)
To write the number 16843009 as hex you would format it as:
01 01 01 01 (1x1, 1x256, 1x65536, 1x16777216)

Here is the basic system for converting a number to hex. example is number 17000000:
17000000 divided by 16777216 = 1.013 ignore the decimals, that gives us 1 so the fourth field is 01
1 x 16777216 = 16777216, 17000000-16777216 = 222784 ok that gives us what is left. Next:

222784 divided by 65536 = 3.399 ignore decimals that gives us 3 so the third field is 03
3 x 65536 = 196608, 222784 - 196608= 26176 ok that gives us what is left to go, next:

26176 divided by 256 = 102.25 ignore decimals that gives us 102 so the second field is 102
102 x 256 = 26112, 26176 - 26112 = 64 ok that gives us the final number, 64

64 divided by 1 equals itself so the first field is 64.

What we have now is: 64 102 03 01, these need to be converted to hexadecimal digits, use a scientific calculator that can convert numbers to hex or run the windows calculator (calc.exe) in scientific mode and convert them by selecting "dec" typing in one number and hitting "hex" writing down the number, selecting dec again and entering the next number, eventually you will have the hex code:
40 66 03 01
(remember to put a leading zero on single digit numbers eg 3 = 03)

As for negative numbers, my own experiments have shown that in some cases, when the number entered is above 2147483647 the resultant number is negative. All higher numbers will be negative numbers, I have not explored negative numbers to much so I am not sure how you format them.

Now needing to go through that system every time I wanted to hack into a game was tiresome and I made a lot of errors so I made a simple program to work this part out. The source code is on page 14 and 15 (near the back of this manual).

How to acquire the necessary information from the game

Since this manual concentrates mainly on the save game component of game hacks, we will need a save game to hack.

Before you can get the information, you need to decide what it is exactly that you wish to change.

Generally it will be the amount of money you have when you load up a previously saved game, but it could be (depending on the game) what your current score is, how much energy the character in the game has, or perhaps what level you start on.

(To keep this simple I will assume that you want to change the amount of money you have in a save game.)

If you have already got a save game, run the game and load it, otherwise load the game and begin a new game, then save it.
Once you have loaded the game, pause it, (if possible) to stop any changes occurring.

Now find the figure that you want to change, (in this case the current balance of money in the game) and write it down. eg. 1000 credits/dollars/pounds/gold etc.

At this point re-save the game, I will tell you why in a second.

Now exit the game to dos (or whatever operating system it is that you use)

Now if you are using dos, while in the directory of the game, type DIR /OD this will give you a directory listing, sorted with the most recent file being shown last. Assuming you keep your computers clock up to date, the most recent file should be the one I told you to re-save above.

If the most recent file does not have todays date, type DATE to see what your computer thinks is the date, and look for any files with this date,

If the date shown is correct then check any sub directories using DIR /OD /S /A /P this will display all files in this directory and all sub directories, including all hidden files and pause at every screen full. Keep an eye on every file's date and time, until you see any with todays date and a recent time.

Write down the file name you find, and the directory that it is stored in.

By now you should have something like this written down:

1000 credits
c:\games\privater\game0003.sav

Now you have acquired the necessary information to hack the save game file you loaded! Simple eh?

What to do with the information from the game

Before we can hack the save game file, we need to know exactly what to look for.

Now would be a good time to type the source code (included before the glossary) into a Pascal compiler and run it, if you have a Pascal compiler that is. You can get Pascal compilers off the Internet or from a computer store.

If you don't have a Pascal editor or compiler, then you will have to do it manually (see the explanation of hexadecimal)

If you have the program, run it.

Enter the number you wrote down, (eg. 1000 in my example earlier) it will then give you some information like:

Enter a number to convert to disk hex: 1000
Original Number entered 1000
base 10 [normal] 232 3 0 0
base 16 [hex] E8-03-00-00
Hit enter to exit

Write down the "base 16 [hex]" information, in the above example it was "E8-03-00-00"

Now would be a good time to decide what to change the figure to, eg. One million (1000000)
Run the program again and type in 1000000, it will give you "40-42-0F-00"

By now you should have something like this written down:

1000 credits
c:\games\privater\game0003.sav
E8-03-00-00
1000000
40-42-0F-00

Now you are ready for the next step.

Actually hacking the game

Cautions: Ok this is the destructive part.
Most of the following instructions can corrupt (stuff up) your save game file, or in very rare cases even make your game trash its own files when you load the save game file.
So before we do anything, make sure that you are not hacking your only copy of the game (if it is shareware or pirated that is).
We should make a backup of the file we are going to modify, the easiest thing would be to insert a floppy disk in a drive, and copy the file we are changing to the floppy disk.
Using my earlier example, I would copy the file "c:\games\privater\game0003.sav" to A: drive, or simply copy it to "c:\games\privater\game0003.**OLD** " .

Lets Begin:
For this part, you will need a hexadecimal or disk editor.

Run the hex editor of your choice, and have it load the save game file you wrote down earlier,
in my example it was "c:\games\privater\game0003.sav".

Select the option which allows you to "search ", It will be called something similar to "find", "search", "advanced search", or "hex search".
When the search panel pops up and asks you to type in the search code, go to the "hex search" etc. part and type in the hex code for the number that you are searching for, in my on going example from the previous section, it would be "E8 03 00 00"

Hit the find / search / ok etc. button on the screen (usually hitting enter will do it) and if the search code is found, it will highlight it on screen, or just move the cursor to it. A hex edit screen looks a little like this:

Offset Hexadecimal ASCII

```
00000070: 00 11 00 07 00 67 00 00 - 00 00 00 27 00 17 C1 5D . . .g.. ...'. -]
00000080: 7F 17 C1 69 66 2E FF 0F - 00 01 01 01 01 01 00 00 □ -if._¤ ........
00000090: 00 E8 03 00 00 A4 A3 00 - 00 00 00 00 00 00 00 00 ._...ñú. ........
000000A0: 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00 ........ ........
000000B0: 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00 ........ ........
```

Go into the hexadecimal area (usually hitting tab, or clicking there with the mouse will take you there) and change the highlighted area to the hex codes of the new figure you chose earlier, in my earlier example it was 1000000 whose hex code is "40 42 0F 00". It would now look something like:

Offset Hexadecimal ASCII

```
00000070: 00 11 00 07 00 67 00 00 - 00 00 00 27 00 17 C1 5D . . .g.. ...'. -]
00000080: 7F 17 C1 69 66 2E FF 0F - 00 01 01 01 01 01 00 00 □ -if._¤ ........
00000090: 00 40 42 0F 00 A4 A3 00 - 00 00 00 00 00 00 00 00 .@B¤.ñú. ........
000000A0: 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00 ........ ........
000000B0: 00 00 00 00 00 00 00 00 - 00 00 00 00 00 00 00 00 ........ ........
```

Now repeat the above bulleted (l) steps until when you enter the search code and start the search, it can't find any occurrences of the hex code you entered.

At this point you simply save changes to the file you are editing and exit to dos, now you run the game, and load the save game file you just edited, if it loads ok and the amount of money has been changed to the figure you wanted it to be, play the game!

Trouble?
If the game didn't load properly, exit to DOS, and copy your backup save game file onto the one you changed. If there was more than one occurrence of the hex code when you changed them only change one of them before saving changes and loading the game.

If it still doesn't work restore the backup save game and change a different occurrence of the hex code.
continue to change different occurrences until when you load the game it works, (or you run out of occurrences, and it doesn't work)

A good way to do this is load the un modified save game, run the search as specified earlier, and put a mark on the paper where you noted the hex codes etc, keep running a search and marking the page until there is no more occurrences of the hex code. eg.

1000 credits
c:\games\privater\game0003.sav
E8-03-00-00
1000000
40-42-0F-00

I I I I I (<- here are the lines)

Then change the first occurrence, run the game, if it doesn't work, restore the save game backup,
cross out the first line, run a search twice then change only the second occurrence, and run the game, keep this up until you either run out of lines or it works.

Possible problems:
When you load the game, it shows you have; -say $100000, but the game keeps telling you that you are out of cash when you try to spend it.

This is an interesting bug I encountered a few times, from what I could tell the game stored the number it showed on screen in one place in the save game, and then stored the amount of money you actually had in a different place!
The way to fix this is to load the save game into the editor and find which occurrence changes your money and which occurrence changes how much money it says you have, and set them to the same number.

When you load the game, the amount of money you have is shown as a negative number, or you are dumped into dos with some sort of overflow error when you load the save game.

This is because you either got greedy and changed your balance to a figure higher than 2147483647 or the maximum figure the game allows, or you simply changed the wrong part of the file, select a lower new balance for your games money, and follow the "Trouble" section above.

List of other games I have hacked

Command and Conquer 1

Capitalism Game (demo) scenario file

Privateer 1

Warcraft 2

HOW to hack these games:

Note: The following sections are for more advanced readers, If you have read the rest of the manual then you most likely qualify as a more advanced reader; but be warned the following is written in a manner which a beginner in computers may have difficulty understanding.

-Consider yourself warned!

-If you wish to try any of the following hacks, and require any help, then ask me, if you don't know me, I can be contacted by Email at hereibe@hotmail.com.

Command and Conquer

Ok on Command and Conquer you can give yourself some extra finances by following the the procedure that follows.

1: When you start a new level, wait for your money to finish counting up, and write down the figure.

2: Save the game (as soon as the money finishes counting), and exit from the program.

3: Execute

hack.pas

and enter the number you wrote down.

4: Write down the XX-XX-XX-XX code you receive.

5: Now type DIR/OD in the c&c directory, the most recent file should be your save game file.

6: Run your hex editor, load the save game file and start a hex search for the number you got from

hack.pas

7: When you find an occurrence of it replace it with something like 00-00-05-00,

8: Keep running a search for the code you got from

hack.pas

and replace all occurrences of it with 00-00-05-00.

00-00-05-00 will give you about 327680.

If you only want a more fair amount of cash to start with, eg. normally you start with 1000 and 5000 would be fairer, then enter the amount you want to get into

hack.pas

just before step 6 and use the hex code you get instead of 00-00-05-00.

Note:

Because of the odd way in which money is formatted in this game, the figure that you changed the money into, may not match what you wanted it to be.

Capitalism Game: Modifying Scenario File

(this works on the demo and may work on the full game as well)
Notes:
Hex Assignments (example in hex editor)

```
00A0 -- -- -- -- -- -- -- --  -- -- -- -- -- -- -- -- ........ ........
00B0 -- -- 01 01 00 00 -- --  -- -- -- -- -- -- -- -- ........ ........
```

FOR "00B2-00B5" TO EQUAL 257 SET IT AS ABOVE.
(THIS EXAMPLE WOULD GIVE YOU 257 YEARS TO ACHIEVE ANY OTHER GOALS SET)
Use ***hack.pas*** to get the correct hex codes for your requirements.

I never figured out where they stored the opening balance when you loaded a scenario.
The scenario I used to work all this out was "Entrepreneurial Spirit". (On the demo of the game)
You couldn't save games in the demo I had, so I suppose you could change your bank balance in the retail version by editing it in the save game.

Offset *What objective it controls*
00B2-00B5 = YEARS TO ACHIEVE GOAL
00B6-00B9 = POINTS AWARDED WHEN GOAL ACHIEVED
00BA-00BD = YOUR PERSONAL WEALTH
00BE-00C1 = NET WORTH OF YOUR CORPORATION
00C2-00C5 = MARKET VALUE OF YOUR CORPORATION
00C6-00C9 = ANNUAL REVENUE OF YOUR CORPORATION
00CA-00CD = ANNUAL PROFIT OF YOUR CORPORATION
00CE-00D1 = NET WORTH OF CORPORATIONS UNDER YOUR CONTROL
00D2-00D5 = MARKET VALUE OF CORPORATIONS UNDER YOUR CONTROL
00D6-00D9 = ANNUAL REVENUE OF CORPORATIONS UNDER YOUR CONTROL
00DA-00DD = ANNUAL OPERATING PROFIT OF COMPANIES UNDER YOUR CONTROL
00DE-00E1 = No. OF COMPANIES UNDER YOUR CONTROL
00E2-00E5 = TOTAL No. OF EMPLOYEES IN YOUR CORPORATION
00E6-00E9 = (%) OPERATING PROFIT MARGIN OF YOUR CORPORATION
00EA-00ED = (%) RETURN ON EQUITY OF YOUR CORPORATION
00EE-00F1 = STOCK PRICE OF YOUR CORPORATION
00F2-00F5 = (%) OWNERSHIP OF YOUR CORPORATION
00F6-00F9 = No. OF PRODUCT CLASSES BEING PRODUCED BY YOUR CORPORATION
00FA-00FD = No. OF PRODUCT TYPES BEING PRODUCED BY YOUR CORPORATION

GOAL TOGGLES (*ON/OFF(01/00*)
019E = MARKET DOMINANCE OF THE AUTOMOBILE INDUSTRY
019F = MARKET DOMINANCE OF THE BEVERAGE INDUSTRY
01A0 = MARKET DOMINANCE OF THE CHEMICAL PRODUCTS INDUSTRY
01A1 = MARKET DOMINANCE OF THE CIGARETTE INDUSTRY
01A2 = MARKET DOMINANCE OF THE COMPUTER INDUSTRY
01A3 = MARKET DOMINANCE OF THE COSMETICS INDUSTRY
01A4 = MARKET DOMINANCE OF THE ELECTRONIC PRODUCTS INDUSTRY
01A5 = MARKET DOMINANCE OF THE FOOD INDUSTRY
01A6 = MARKET DOMINANCE OF THE FURNITURE INDUSTRY
01A7 = MARKET DOMINANCE OF THE JEWELRY INDUSTRY
01A8 = MARKET DOMINANCE OF THE LIVESTOCK PRODUCTS INDUSTRY
01AF = MARKET DOMINANCE OF THE TOY INDUSTRY
01B0 = MARKET DOMINANCE OF THE WATCH INDUSTRY
01B1 = MARKET DOMINANCE OF THE APPAREL, FOOTWEAR & BAG INDUSTRY

Privateer 1

- To change your amount of credits,
- Simply write down how may credits you have,
- Save the game, exit privateer,
- run hack.pas enter the amount of credits,
- write down the hex code you get,
- load your save game file into your hex editor,
- search for the hex code you got,
- change it to 00-00-00-01
(this will give you a little over 16 million credits!),
- save the modified file, exit,
- load privateer,
- load your save game,
- PLAY!

The amount of money you will have will be more than enough to buy the best ship, deck it out with the best weapons, shields and equipment. And even for a while leave you enough money to buy a new ship every time the old one is damaged!

Warcraft 2

To hack this game is the same as hacking command and conquer.

I would wait until later levels before changing this one, so that you have a more definate original hex code, as there may already be other data in the file with the same hex code. eg. level 1, there is about 20 occurances of the hex code you need to search for.

BUT, if you change any areas which are set to
01-00-01-01-01-00
It will change the computer players AI (Artificial Intelligence attack strategies) and it is even possible by changing the correct "01" areas to disable a unit type from attacking!
It will just stand there even when one of your units are standing next to it!

To toggle an AI zone change the "01" (smile face character) to a "00" (nul character).

Experiment for your self, just don't change anything but the gold, and groups of the following type hex codes eg:

01-00-01-01 01-01-01-01 - 01-01-01-01 01-01-00-01
01-01-01-01 01-01-01-01 - 01-01-01-01 01-01-01-01
01-01-01-01 01-01-01-01 - 01-01-01-01 01-01-00-01

You can spot the AI control areas by large groups of the smiley face character in the hex editor. Their position varied, so i can't give a specific offset.

I never got around to mapping these. But I got the impression this gave you a much more extensive control of the games AI settings than even the level editor offered.

Note: 00(off) and 01 (on)

Source code for hack.pas

```
program hexhack;
{This program was designed for Turbo Pascal, if it doesn't}
{work on your version of pascal, remove the uses crt; and clrscr;
commands}
{Program to convert a number into it's file hexadecimal equivalent}

uses crt;

var
number, {Number user enters}
n1, {hex number range 1 (0-255)}
n2, {hex number range 2 (256-65280)}
n3, {hex number range 3 (65536-16711680)}
n4, {hex number range 4 (16777216-4278190080)}
n1a, {n1 remainder}
n2a, {n2 remainder}
n3a, {n3 remainder}
n4a, {n4 remainder}
num, {number for procedure hexit to convert}
sixteen10, {the left hex char}
lessixteen:longint; {the right hex char}
hx:string[2]; {the total hex char}

procedure hexit;
{procedure to convert the number in num into hex format}
begin
hx:='';
sixteen10:=num div 16; {seperate the base 16 "10"s field}
lessixteen:=num mod 16; {seperate the base 16 "1"'s field}

{convert the base 16 "10"s field into hex character}
case sixteen10 of
0 : hx:='0';
1 : hx:='1';
2 : hx:='2';
3 : hx:='3';
4 : hx:='4';
5 : hx:='5';
6 : hx:='6';
7 : hx:='7';
8 : hx:='8';
9 : hx:='9';
10 : hx:='A';
11 : hx:='B';
12 : hx:='C';
13 : hx:='D';
14 : hx:='E';
15 : hx:='F';
else
end;
```

```
{convert the base 16 "1"s field into hex character and combine with "10"s}
case lessixteen of
0 : hx:=hx+'0';
1 : hx:=hx+'1';
2 : hx:=hx+'2';
3 : hx:=hx+'3';
4 : hx:=hx+'4';
5 : hx:=hx+'5';
6 : hx:=hx+'6';
7 : hx:=hx+'7';
8 : hx:=hx+'8';
9 : hx:=hx+'9';
10 : hx:=hx+'A';
11 : hx:=hx+'B';
12 : hx:=hx+'C';
13 : hx:=hx+'D';
14 : hx:=hx+'E';
15 : hx:=hx+'F';
else
end;
write (hx); {display the hex code converted from num}
end;


begin {the start of the program}
clrscr;
write('Enter a number to convert to disk hex: ');
readln(number);

{do the main calculations}
n4:=number div 16777216;
n4a:=number mod 16777216;
n3:=n4a div 65536;
n3a:=n4a mod 65536;
n2:=n3a div 256;
n2a:=n3a mod 256;
n1:=n2a;

{display results, have them converted to hex, and display in hex}
writeln('Original Number entered ',number);
writeln('base 10 [normal] ',n1:4,n2:4,n3:4,n4:4);
write('base 16 [hex] ');
num:=n1; {set what needs to be shown as hex}
hexit; {convert num to hex and show it}
write('-'); {display a seperator character}
num:=n2; {set what needs to be shown as hex}
hexit; {convert num to hex and show it}
write('-'); {display a seperator character}
num:=n3; {set what needs to be shown as hex}
hexit; {convert num to hex and show it}
write('-'); {display a seperator character}
num:=n4; {set what needs to be shown as hex}
hexit; {convert num to hex and show it}

writeln; writeln('Hit enter to exit');
readln;
end.
```

-- End of source code -- >>>>

Glossary

Disk editor , A computer program used to edit directories, files, and whole disks.

Byte , The computer space occupied by a single text character.

Character , A letter, number, or punctuation mark etc.

Hack , 1: (c1978-present day) A sequence of steps or a program which does a sequence of steps for changing a computer file or program to accomplish some objective, 2: (c1985-present day) To attempt to gain access to files on a computer stored elsewhere via a network connection, 3: (c1940-1976) Someone who thinks they know how to do something but don't / unintelligent fool, 4: to wreck / to chop.

Hacking , 1: (c1973-present day)The action of changing a file or program to accomplish some objective. 2: (c1973-1985) "Hacking together some code" System administrators would "hack together" a piece of code to fix/patch a problem on the computer system. 3: (c1985-present day) Attempting to gain access to files stored on a computer located elsewhere via the manipulation of a computer network. 4: wrecking/trashing/chopping up something,

Hex , 1: Abbreviation of hexadecimal, in computing terms refers to binary/machine code etc. formatted as hexademial numbers to minimise space used up on screen. 2: A curse or spell cast on someone.

Hex editor , A program which converts text/extended/ctrl codes in a computer file into hexadecimal format, and allows you to edit the byte/code of each character. Disk editors usually have a Hex editor built-in to it.

Integer , A number with the decimal places removed.

Offset, A counter which counts in hexadecimal the amount of characters from the start of the file.

Save game file , A file placed on the disk by a game when you wish to stop playing the game, but continue where you left off at a later time.

Scenario file , A preset set of objectives, conditions and settings which the person needs to beat in order to win. Usually used when refering to computer games.