Glory of Heracles: The Labors of the Divine Hero Save State Hacking Guide

by MotherKojiro Updated to v1.0 on Nov 5, 2018

```
Heracles no Eikou:
A Save State Hacking Guide
By: Alice Kojiro (MotherKojiro(AT)gmail(DOT)com)

Note: This work is mine exclusively.  I've spent several hours writing it.  If
you don't believe me, try writing one yourself.  Anyway, if you need to contact
me, just send me an e-mail.  E-mails have 3 main criteria.  First of all, they
need a subject line that tells me that the message is about my walkthrough.
Another important criterion is that it needs to be suggestions about the
walkthrough.  I don't appreciate invitations to blogs, invites to chat rooms,
flaming, or anything like that.  Your suggestions can be anything from spelling
and grammatical errors to new sections you think I should add.  The only other
reason you should be writing to me is to ask me to use this guide on your
website.  You cannot do so without my consent, as it is illegal.  All you need
to do is ask and it is highly unlikely that I will deny you the use of this
guide.  You also have to give me credit for writing this.  Failure to do so is
plagiarism.  I appreciate your cooperation.  Thanks for choosing this guide.
                                                                        ~Kojiro

=======
History
=======

Version 1.0 - A list of relevant addresses and offsets.


=================
Table of Contents
=================
I.     Intro
II.    Thanks
III.   An Introduction to Hex Editing
IV.    Addresses
V.     Offsets


========
I. Intro
========
Heracles no Eikou isn't a great game.  It's slow, clunky, and unforgiving even
by the standards of the original Dragon Warrior.  The graphics range from
impressively bizarre enemy design (in a good way) to eye-bleeding interior
decorating (in a bad way), and the music screams low budget.  Story!?  Not in
this era, though it does a pretty decent job of more-or-less accurately
representing Greek Mythology; this isn't Valkyrie Profile.  The game is so
grueling, in fact, that I made this guide from scratch just to drag myself
through it; I've seen the later games in the series, and they look quite good.

And yet, there is an enchanting low-budget charm to it all.  Walking into one
of those hideous dwellings, with its vaguely warm music blaring through my
speakers, it felt like home in a way I can't adequately describe.  It reminded
me of cloudy Sunday Mornings at home, decades ago, sitting in front of the TV
```

with my family, watching MASH or playing a few rounds of Ghostbusters or Three Stooges on the NES, or Thunder Blade on the SMS. Is it the music? The visuals? The low-budget charm? I don't know, but there's something about it that I remember very fondly, and if you know me, you know that I'm not one for nostalgia.

Maybe you want to play the later games in the series, and just want to blow through this first title. Maybe you've found a similar charm, but just can't take what this game throws at you. Whatever the case, I'm here to make your life a little easier. I put this together as I went along, and filled in the blanks at the end, so most everything has been tested in FCEUXDSP 1.06. It is my sincere hope that you can benefit from my labor.

==========
II. Thanks
==========
Nintendo for making the Nintendo Entertainment System.
Data East for making this game.
My lady for putting up with me.
Mom, Dad, and the cloudy Sunday Mornings we used to spend together.
GameFAQs for hosting this guide.
You, the reader.

==================================
III. An Introduction to Hex Editing
==================================
For the uninitiated, this is a guide for editing "save states"; I put this in quotes, because "save state" is a function. You are saving the state of the game; what you are making is a saved state. Semantics aside, it's like a snapshot of the game, not just visually, but with all of the data that's in the RAM at that time. So, it would include your current position in the game, your life, ammo/MP, money, etc. It started out as a handy feature in emulators, but now, most new ports of old games have this feature.

Imagine, though, being able to crack that open and tweak it to give yourself unimaginable power, or even just a little boost. Oldschool cheat devices, like Game Genie and Action Replay used to force certain values to stay at a certain number, but did a similar thing. This is a little more hands-on, but if you know what you're doing, you can cause a lot of the same effects, some even better. In fact, through this process, I once made an invincibility code for Kid Icarus on the NES, even though no such Game Genie Code is known to exist. One wrong move and you can really screw things up, but find the right thing, and you can do something truly revolutionary. Just make sure to back up your files before experimenting with them.

The first thing that you will need is a hex editor. You can find these fairly easily; the one that I prefer, and have used for years is XVI32, which is free and has a lot of handy features. What you want to open with this program is the actual state that you've saved; for FCEUXDSP users, the file extension will be .fc?, where ? is any number from 0-9. This will show you all of the data currently being used by your game, and you can do what you like with it. But what are you even looking at?

Electronics, at their very core, operate on a system known as binary, which is a number system that has only zeroes and ones. Our own number system is called decimal, which has ten digits: 0-9. So, like how we have the ones place, tens place, hundreds place, and so on; binary has the ones place, twos place, fours place, and so on. Like with decimal, once a place gets

its highest value, any increase moves on to the next place, but in binary, there are only 0 and 1, so this happens much more quickly. For example, 10 in decimal is ten, because it is one more than nine, its highest numeral; in binary, 10 is two, because it is one more than one, its highest numeral.

So what is hexadecimal? Binary is base two, and decimal is base ten; hexadecimal is base sixteen. Why sixteen? For one thing, binary has a sixteens place; 10000 in binary is sixteen. For another, a common measurement of computer space is a byte, which is eight bits. A bit is a binary one or zero, so a byte has a maximum value of 11111111, which is two hundred fifty-five; you've likely seen this number in games before, and this is why. In hexadecimal, which uses digits 0-9 followed by A-F, this value is FF. Simply put, hexadecimal does an excellent job of being binary shorthand.

Much of what you can do with hacking states involves numbers, such as your HP, MP, stats, experience, money, etc. This is pretty straightforward, but there are a few common tricks that might throw you off. The biggest thing to get used to is that values greater than one byte are often what is called big-endian. Big-endian means that the bytes are in reverse order. Say, for instance, that you want one shy of ten million gold, which is a common money cap in JRPGs. In hexadecimal, that value is 98967F, but since it's big-endian when stored in the state, you would enter the byte values in reverse order: 7F 96 98. Almost any time that you have multiple bytes for one value, it will be like this.

"But this is hard; how do I convert decimal into hexadecimal?" Fear not; you can find scientific calulators that will convert the values for you, both to and from hexadecimal. In fact, most also have binary and octal (base eight). If you don't have one, and don't want to buy one, there's likely one on your computer. If your version of Windows has a calculator with a Programmer setting, then use that; older versions should have conversions under Scientific. I haven't used Mac or Unix in ages, and I don't have a tablet or smart phone (yes, really), so your mileage may vary with these.

Save hacking can do more, however, such as alter your inventory and equipment. It's a similar process, but each weapon is represented by a value; the system knows to interpret this value not as a number, but as the representation of an object in the game's code. If you want a wooden shield, which is represented by the value 0E, all that you would need to do is change an empty inventory slot's value to 0E, save the file, and load the state. If you have a wooden shield and would like for it to be a copper shield (value 0F), you can also just change the existing 0E value to 0F, and do the same. In some cases, you will also have a separate byte dictating the quantity of said item, which may either be right next to the item, or in a different place entirely. Of course, if you do not change this quantity from zero, your item may not even show up. This particular game does not have item quantities, so that is not a worry here.

Finally, a few words of caution. While you can directly alter a character's equipment or stats, it is generally best to let them change naturally. Simply changing values like these will work with some games, but in others, they can reset on you or even just act like they were never changed. For example, if you want better stats, the best thing to do is to increase your experience and level naturally. There are a few games where you will need to do this level by level; fortunately, Heracles no Eikou is not one of which. For equipment, it's best to either give yourself the gold to buy it if you can, and if you can't, to add it into your inventory and equip it, so that the game recognizes the equipment change in all areas.

=============
IV. Addresses
=============
Here we are at the main course of the guide. Values are big-endian, as usual, though that only really applies to experience and money. The three weapon slots are a little screwy in that you're only really using the one in the first slot; the other 2 are their own special inventory space for other weapons, since they do not appear in the menu with your items. Because of that, DO NOT put weapons or armor into your item slots. It shouldn't destroy your game, but I'm pretty sure that you won't be able to use them in there. I mean, go ahead and try to stab your enemies with a pork chop if you really want, but do so knowing that you risk more than just your own amusement. Maxing out your experience from the get-go doesn't cause any adverse effects that I noticed; I have used it to increase my level by several all at once, and my stats wound up right where they should have been. The only other thing of note is that medicine doesn't exactly have a slot in your inventory; it's more hard-coded into a floating slot at the beginning. If you want medicine, just increase the quantity at hex address D8; the natural cap is 6. Take note that if you are not using FCEUDXSP version 1.06, the addresses may be different; I've encountered this before, but the values and offsets will be the same, so you can just search for them to figure out where everything is for you.

CA-CB - Experience
C6 - Current HP
C7 - Strength
C8 - Defense
C9 - Intellect
D1-D2 - Gold
D3 - Equipment Slot 1 (Weapon)
D4 - Equipment Slot 2 (Weapon)
D5 - Equipment Slot 3 (Weapon)
D6 - Equipment Slot 4 (Shield)
D7 - Equipment Slot 5 (Armor)
CC - Durability Slot 1 (Weapon)
CD - Durability Slot 2 (Weapon)
CE - Durability Slot 3 (Weapon)
CF - Durability Slot 4 (Shield)
D0 - Durability Slot 5 (Armor)
D8 - Medicine Quantity
D9-E1 - Item Slots

==========
V. Offsets
==========
These are the values that you would enter for equipment or item slots and what they represent. If it isn't obvious, 00 is a blank space, 01-0D are weapons, 0E-15 are shields, 16-1C are body armor, and the rest are items, both key and consumable. 45-4B are the names of the towns, and I'm not sure what their purpose is; my best guess is that this is for the data for the Travel Wings, but I couldn't find anything in the state's data to show where that would be stored. 4C and 4D are the inn and bar, respectively, and I'm not sure what these do, either. The obvious answer would be that you can use an inn or bar anywhere, or even that they would automatically heal you or tell you how much experience you need to reach the next level after battle, but they don't do either of these things. They can't be used in dungeons, out of dungeons, or in battle, though they can be sold. 4E-FF are blank spaces; some say things when you use them, so they might be NPCs. Some will cause your item menu to not even open, so use them with caution. My theory is that instead of separating all of these wildly diffent things into different categories, the

programmers decided to just lump them all together, but I've experimented with hex editing for tons of JRPGs on the NES and other systems, and I've never seen anything like this before.

01 - Knife
02 - Copper Sword
03 - Iron Sword
04 - Silver Axe
05 - Hammer
06 - Steel Sword
07 - Trident
08 - Ares's Axe
09 - Atlas's Spear
0A - Baneslayer
0B - Evil Halberd
0C - Zeus's Sword
0D - Silver Bow
0E - Wooden Shield
0F - Copper Shield
10 - Iron Shield
11 - Steel Shield
12 - Holy Shield
13 - Caesar's Shield
14 - Rusty Shield
15 - Zeus's Shield
16 - Leather Armor
17 - Chain Mail
18 - Copper Armor
19 - Iron Armor
1A - Steel Armor
1B - Magic Armor
1C - Zeus's Armor
1D - Medicine
1E - Key
1F - Mirror of Etna
20 - Water of Life
21 - Silk Reins
22 - Stone Tablet
23 - Spring Drop
24 - Carrot
25 - Candle
26 - Mind Window
27 - Ice Bowl
28 - Utopia Cane
29 - Bone
2A - Chicken
2B - Pork Chop
2C - Fire Pillar Balls
2D - Spring Drop
2E - Thunder Cane
2F - Blacksmith
30 - Dream Flute
31 - Compass
32 - Bolk's Venom
33 - Travel Wings
34 - Ship
35 - Lamp
36 - Rainbow Ring
37 - Holy Bell
38 - Heal Seed

```
39 - Night Curtain
3A - Crystal
3B - Sea Lyre
3C - Prawns
3D - Pegasus
3E - Leila's Jade
3F - Gold Horn
40 - Snapper
41 - Mackerel
42 - Cross
43 - Radish
44 - Cabbage
45 - Athens
46 - Pella
47 - Selene
48 - Hebe
49 - Nana
4A - Tartar
4B - Leaneira
4C - Inn
4D - Bar
4E-FF - ????????
```