

Branching

This section contains the most technical aspects of event hacking. So far, the game has executed our event scripts in a straight line. If the beginning of the event was at address CA/2000, the game would execute the byte at CA/2000, then the byte at CA/2001, then the byte at CA/2002, and so on. But sometimes, we may want the event to “jump” or “branch” from CA/2000 to CA/4000 (or some other address). There are several reasons why you may want to do this:

- To reuse a part of the event script several times. For example, when the party rests at an Inn, the game will always fade out the screen, play the “Nighty Night” song, restore the party's HP, and fade in the screen. It would be wasteful to write “96 F2 30 31 05 09 E0 06 27 FF F0 B8 B2 BD CF 00 FA F3 10 31 82 09 FF 94 B3 03 FE C7 00 FE” inside every cutscene triggered by speaking to an innkeeper. Instead, the developers placed that lengthy script at CA/00EA, and jumped to this address in every Inn cutscene.
- To skip over part of an event. If Cyan is in the party at the Imperial banquet, he will say an extra line of dialogue about General Leo. If he is not in the party, this part of the event must be skipped by the game.
- To use extra space. If you expand the size of your ROM using FF3usME, you will need to jump to the additional space to use it.
- To repeat part of an event many times (perhaps an infinite number of times).

Jumping to a Subroutine

The simplest way to jump to another address is to use the B2 command. Let's say that we want to jump to CA/4020. Following the instructions in the Event Commands Document, we know to start by subtracting CA0000 from CA4020, giving us 00 40 20. Then we need to reverse the order of the bytes. The full command is B2 20 40 00. You will need to follow this process often when jumping/branching to an address.

When the game jumps to CA/4020, it will execute the event script found there, then return once it reaches an \$FE command. The important thing to note is that the \$FE command does not end the event if it is executed inside a subroutine. Instead, it will return to the address at which you left off. If the event script is 4B 23 01 **B2 20 40 00** 4B 23 02, the game will display the first text box, execute the subroutine at B2 20 40 00, then display the second text box after returning from the subroutine. If you wanted the event to end directly after executing the subroutine, you could have written 4B 23 01 **B2 20 40 00** FE instead.

This command alone allows us to reuse parts of the event script, as in the innkeeper example above. It also lets us make use of extra space from expanding the ROM. Do not use this command to infinitely repeat part of an event; if you jump to subroutines 255 times without returning, the game will crash.

Event Bits

A way to branch based on certain conditions would also be very helpful. For example, after the first set of guards in Narshe is defeated, the player should not be able to encounter them again. To accomplish this, we need an “event bit”. Every event bit contains a single value that can be either 1

or 0. At the beginning of the game, the event bits are all 0. After you fight the first set of guards in Narshe, an event bit is changed to 1. This is how the game remembers that you have fought this set of guards.

Every time you step on that tile, the game will check to see if that particular event bit is 1 or 0. If it is 0, the game knows that the player has not fought the guards yet. If it is 1, the player must have fought the guards already. Even if the player saves and reloads the game, this event bit will revert to the state that it was in when the game was saved.

If we use event bits correctly, the game can answer questions like “Has the party met Shadow yet?”, or “Has the party defeated my optional superboss?”. We have a lot more power than simply preventing a cutscene from repeating itself.

From:
<https://www.ff6hacking.com/wiki/> - ff6hacking.com wiki

Permanent link:
<https://www.ff6hacking.com/wiki/doku.php?id=ff3:ff3us:tutorial:events:branch&rev=1525128574>

Last update: **2019/02/12 09:54**

