

Members **0** 1 26 posts Updates since the second release:

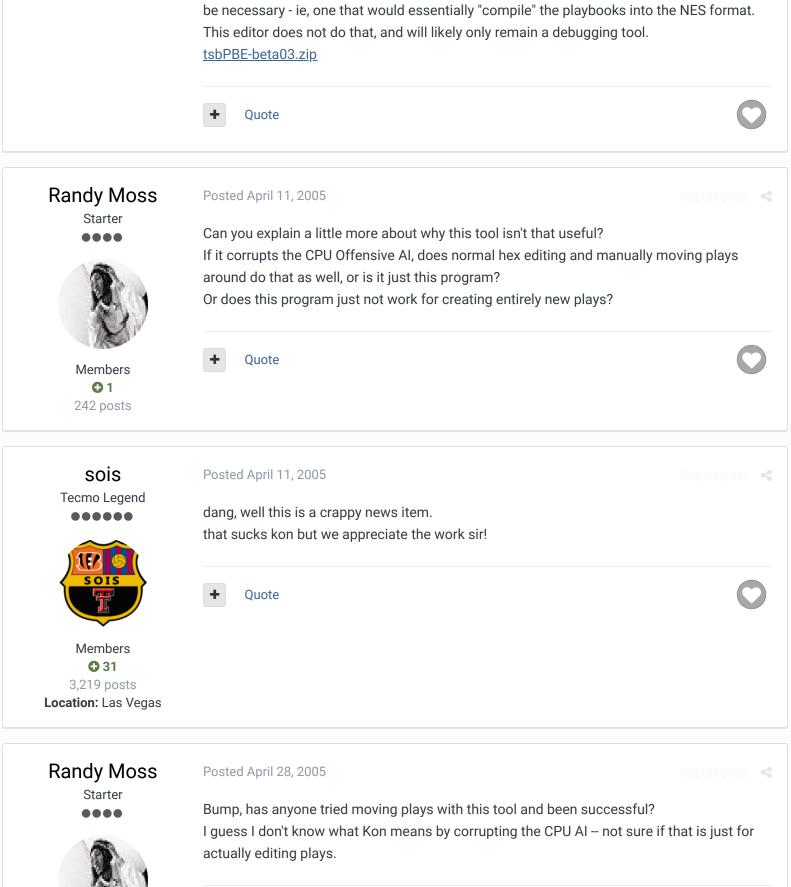
- * BUGFIX: blitz byte (hopefully) saved correctly
- * Added more information to various command codes
- * Sequence data is reordered upon saving (*Nukes CPU AI*)
- * Added "Safe Save" option

The purpose of this release is to correct a problem that was brought to my attention regarding the blitz byte not being saved properly. It *should* be resolved now, but you'll have to see for yourself if it works.

Other than that, there is **no additional useful functionality**. The editor does load the squences more accurately than Beta 2, but this doesn't translate into any practical applications.

If you want to use the editor to swap plays around, use the *Safe Save* option. It only saves the play slot information. Doing a *Full Save* will damage the ROM in that all offensive CPU AI will be corrupted.

The discovery that TSB stores CPU play logic within the sequences threw a monkey wrench into the route I was taking. To make a useful play editor, a completely high level editor would



Members **0** 1 242 posts Quote





Members 1 26 posts

Posted May 19, 2005

I understand how to make user defined plays, all the way down to the routes, etc. There is still some unknown information, but it's nothing that couldn't be figured out. The kicker is that embedded in the route information is the CPU logic. The CPU is not smart, it just does what the logic tells it to *per route*. In other words, if you create a play you also have to tell the CPU how to use the play.

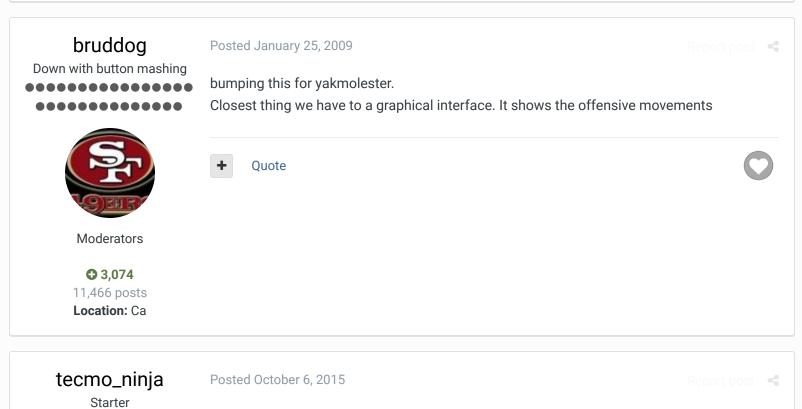
What happens in the play sequence is at the end, a "take control" byte command is issued. If you are a human then you can now control the player and the play sequence is halted. However, if you are a CPU, the play sequence continues according to the logic that immediately follows it. (The last beta of my editor did not even attempt to load CPU data and thus, when saving the playbook, all CPU logic is overwritten with garbage.) The logic for a QB might look like:

- 1) Drop back for pass.
- 2) If random number is A then pass to a player
- 3) If random number is B then move to a location (scramble)
- 4) While scrambling, if random number is C then pass to a player
- 5) Run with ball.

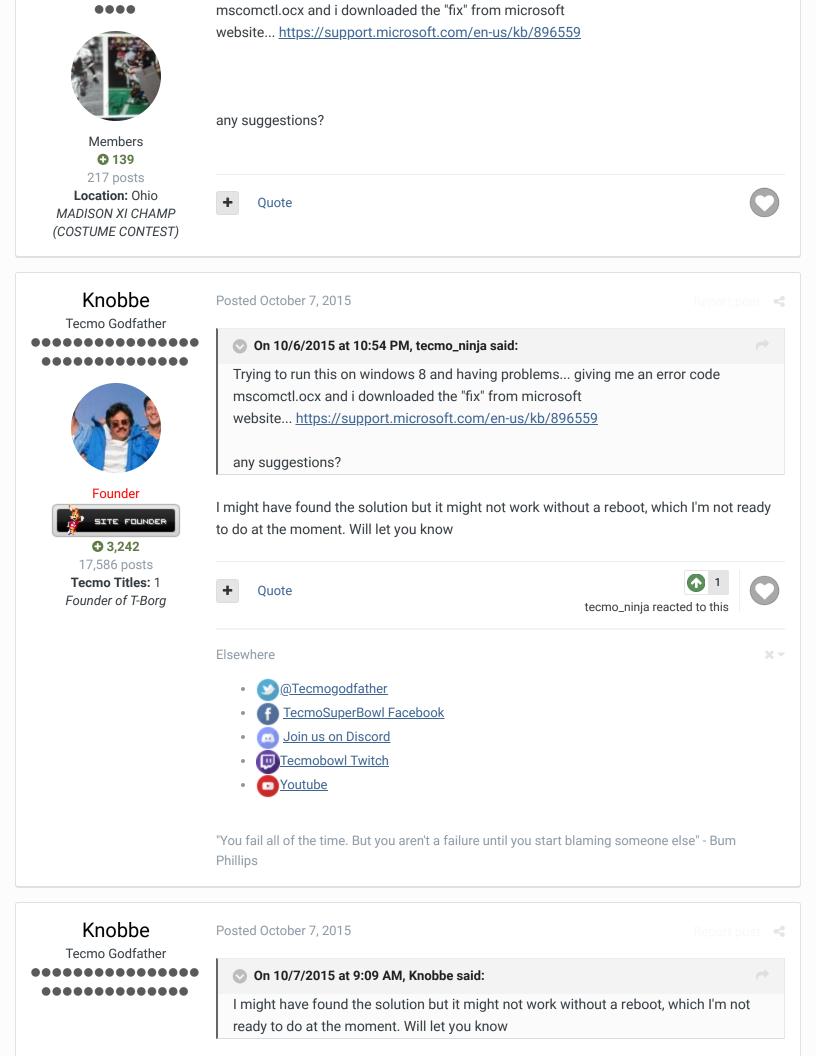
I never had time to completely look into this, but to me it looked like the best way to create a usable playbook editor would be to actually create a program that lets you draw plays up, and then press a "compile" button that creates the NES byte code needed for TSB. In other words the program would not be able to extract information from the TSB ROM, it would only be able to export a playbook to it.

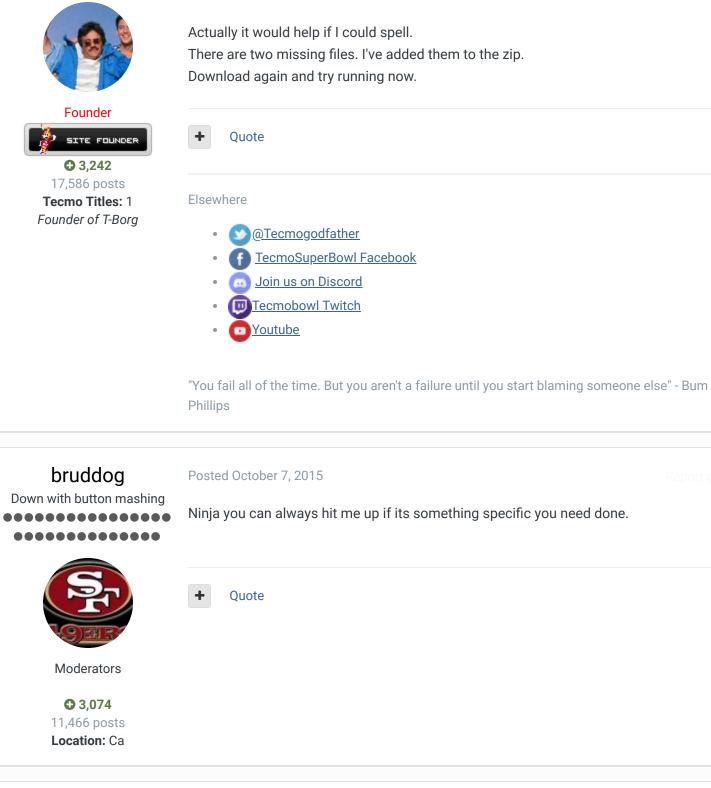
The method my playbook "editor" uses would have only worked if things were a bit more structured in the TSB ROM.

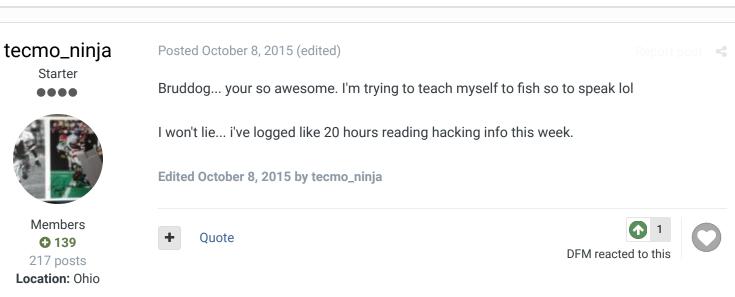


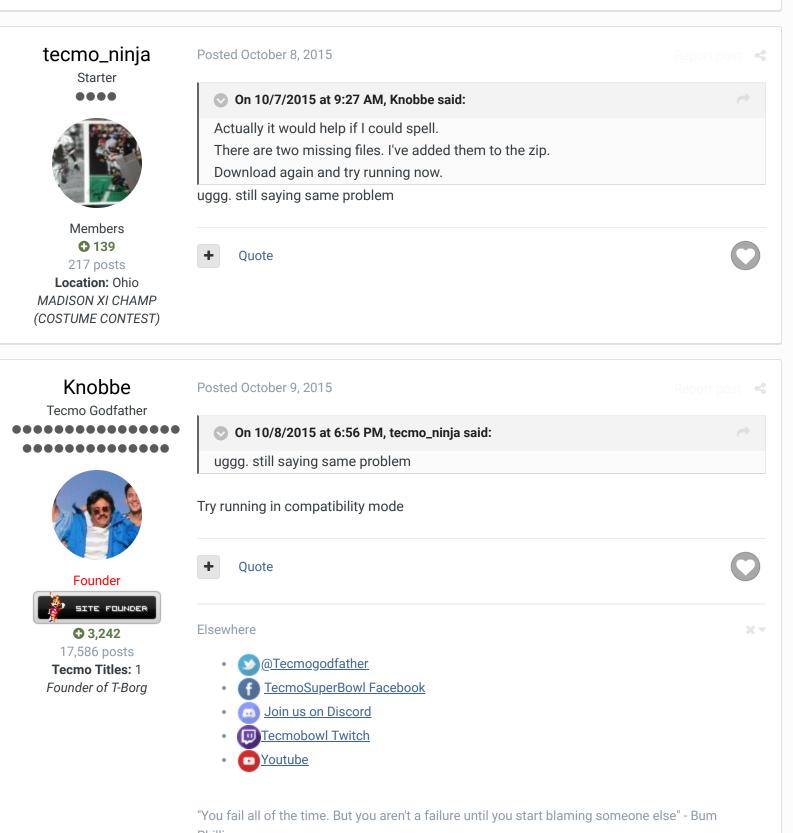


Trying to run this on windows 8 and having problems... giving me an error code









Phillips





RECENTLY BROWSING 1 MEMBER

SBlueman

Home > Forums > Hacking/Emulation > Editors/Emulation > NES TSB Play Editor BETA3 release

✓ Mark site read

Theme ▼ Contact Us

TecmoBowl.org Powered by Invision Community