

NavAdder

aka Navigation Attacher and Tweaker
- a tool for UT'99 using XC_Engine -
Update 1 notes

Purpose:

This is a mutator - a very small one working with "body parts". This means it will be loaded and will check map. If desired package for map is found it will load that package, an optional module, and launching it. That module-package will execute tweaking operations in map. To summarize, we will not have an almighty package-mutator knowing a lot of maps and testing things in each of them. It will be a small mutator checking if a tweaker file is found and letting tweaker to do its job accordingly and staying passive - all tweaks are done once or, depending on situation some latent stuff is injected being more or less active, triggerable, etc.

Mutator operation:

NavAdder.NavAdder (aka XC Navigation Adder - in mutators menu) is loaded as server-side, no packages. It is dependent on XC_Engine so we won't mess with packages at this point. Mutator will print something in logs, files <name.class> which is looking for. If mutator will find these 2 files it will load them. First file is also a server-side which I'm using to patch things without packages. The secondary will have player needs so mutator will put it automatically in "PackagesMap", no ini editing is required at this point. Second file being a package should have the name with prefix "P_" (no quotes). If game/server uses hooks for teleporters this mutator has to be prioritized to do the job first, otherwise results are crapped up.

What happens in fact ? First module loaded addressing authority - server, will be capable of tweaking paths, if map is borked in some areas it will inject even a fake navigation. It will correct movers, blocking bad spots where Bots are acting nasty, module was properly tweaking levels where MBot (produced by self person) works like a charm. This server-side actor can be modified anytime because it's not for players - so to speak, we can develop another paths Network whatever next week. Modules are not dependent of main mutator as main mutator has no relations with module, it is The LOADER and nothing more.

The second module is addressing to audio-visual stuff, tiny borks can be covered with decorations and this package is mapped automatically so player will have it without to modify INI files for each map.

Each map which is fixable in run-time can be a subject for this mutator. A certain package is doable anytime so to speak mod is supporting UPDATES related to server-side parts and new maps.

Explained files:

NavAdder.u - is the main mutator that needs to be loaded;

NavAdder.int - file opening the interface in Mutators list - Off-Line play;

NavAdder.ini - file having a configurable array with map-names that will be skipped from patch seeking task - **REASON for update 1.**

Update 1 notes:

Reason for updating has been initialized since I could see patch files loaded for a map which was not target for NavAdder because it was wearing the same group of letters in map-name. It was about patch Files for map **MH-CrystalMine3** loaded even in Map **MH-UM-CrystalMine3** because of string "**CrystalMine3**" used as target package. I'm not against patching the second map, but this task would be impossible due to the same name used by patch files probably "new map" requiring additional tweaking in new files.

Tech explained: Problem is debated different right now, NavAdder will scan for a file matching map-name. If is/are found, it/they will be loaded, if not, NavAdder will scan for alternate old method, truncated map-name, if is found one or both patch modules, they are loaded. If a long named modified map, like in case above, doesn't have a patch but we want old original map patched, we can except long-named map from being patched with original files by listing it in INI file without extension. All maps listed there (1024 places) will except NavAdder from patching them. Changes to INI file are doable even if server is running. NavAdder will ONLY read INI content and not saving it - preventing to write-access the drive at each map change.