
Subject: : Open Amiga

Topic: : BOOPSI Menu Class: first update

Re: BOOPSI Menu Class: first update

Author: : trixie

Date: : 2014/7/5 11:53:12

URL:

@Chris

Quote:

If you do it via NewMenu you can do other funky stuff, like menu callbacks, which is probably what you want in a OO based system.

I quite like the idea that each menu object can be associated with a callback function that gets invoked automatically upon the user selecting the respective item. But I wouldn't like to tie this to either NewMenu, or Window Class.

So I have taken my own way. You can now associate a "handler" function with any menu object, which you pass via MENU_Handler. This tag takes a pointer to a hook function, where the particular reaction to selecting the item will be handled. The hook for this hook function is allocated, installed and disposed of internally by the class. (This of course saves a lot of work. Considering that menus can contain dozens of items, just imagine having to prepare dozens of hooks manually!)

Upon receiving a menu event for your window, you just call the MM_HANDLEINPUT method and that is all. The method will process the menu selection and call the respective handler. (The ARexx class works in this way, too, so I'm not introducing any foreign solution here.)

If we manage to establish some cooperation between Window Class and Menu Class, the MM_HANDLEINPUT method could even be called by Window Class upon receiving IDCMP_MENUPICK. Thus, the programmer would become completely free of having to process menu events in the event loop.

Along with MENU_Handler I have also added MENU_UserData, through which you can associate arbitrary data with the menu object. If MENU_UserData is provided, the hook for the handler function will put it in its ASOHOOK_Data, so that the user data can be accessed from within the handler if needed.