

---

Subject: : Open Amiga

Topic: : BOOPSI Menu Class: first update

Re: BOOPSI Menu Class: first update

Author: : Chris

Date: : 2014/7/13 11:09:54

URL:

@trixie

Quote:

1. The low-level stuff (like allocating/freeing hooks) is handled by the class. BOOPSI is the highest-level component of Intuition so it had better behave like one; and as I've already said above, I can't imagine having to manage two dozen hooks on top of two dozen hook functions.

Agreed, allocating hooks manually would be annoying. `window.class` already solves this one - you just pass the function pointer.

Quote:

2. The callback function is a function that I need to write anyway - for example, an `about()` function that creates and displays the info requester - and which I can reuse and invoke from other parts of the program (such as the toolbar).

And there comes my current problem. If the `about()` callback is written as a hook function (i.e. with parameters required by the AmigaOS hook API) and is invoked by the class (which duly supplies the hook and all), how would I invoke the callback from the toolbar when there's no hook to pass as parameter?

Pass NULL? It's a fair point, although how often are toolbar buttons duplicates of what is in the menu?

Even if your hook functions just call a function elsewhere it's still easier to manage.

Quote:

The ARexx Class uses callbacks but doesn't use hook functions. This is something I'd like to know more about.

That may be a better option, although you are still constrained to the parameters for the callback that the class wants you to use.

arexx.class uses:

```
STATIC VOID rx_func(struct ARexxCmd *cmd, struct RexxMsg *rxm __attribute__((unused)))
```

Is this really any better than:

```
static void menu_func(struct Hook *hook, APTR window, struct IntuiMessage *msg)?
```