
Piece-Menus

By: D. Austin Henderson, Jr. (AHenderson.pa@Xerox.COM)

INTRODUCTION

This module provides two solutions to the problem of menus with too many items. Which is useful will depend on the inherent structure of the items. 1) CHUNK-MENUS: For use in which the items have the simple structure of one long list: Break the items up into chunks following the ordering of the items in the list, and then provide a menu which presents one of those pieces and a way of getting other menus containing the other pieces. 2) KEYWORD-MENUS: For use when it is possible to associate keywords with the items: Break the items up into the sets which share the same keyword, and then provide a menu which presents one of those chunks and a way of getting other menus with other keywords. As with standard Interlisp-D menus, these specialized menus are created by a "create" function (cf. (CREATE MENU)), and are used with a single "invoke" function (cf. (MENU menu)). The items and other information are as with standard menus, except that the other information is presented to the create function in the form of a Plist.

CHUNK MENU

A Chunk Menu appears as a single menu, but is really a data structure encompassing a number of pieces, each represented by a menu, between which the operator can move to find the item desired. Each piece is in three parts: a set of "required items" which will appear in all chunks, a set of indicators for all the chunks, and the items in this piece. Each chunk has up to 30 items in it (it can be varied using the CHUNK.COUNT property). If there are fewer than one chunk's worth of items, the CHUNK-MENU behaves just like a standard menu.

(CHUNK.MENU.CREATE *ITEMS PROPERTIES REQUIRED.ITEMS*) [Function]

Creates and returns a Chunk Menu, a data structure of menus each containing some of the (presumably large number of) items.. All items are as with standard menus. The properties understood are: CHUNK.COUNT (see above), TITLE, CENTERFLG, MENUFONT, ITEMWIDTH, ITEMHEIGHT, MENUBORDERSIZE, and MENULAYOUTSIZE. The actual menus are created only when needed.

(CHUNK.MENU.INVOKE *CHUNK.MENU POSITION*) [Function]

Carries out the interaction with the user to select an item from a Chunk Menu. If *POSITION* is non-NIL, the menu will appear at that position, otherwise it will appear under the mouse. All interactions are as with the standard menus. Returns the value produced when a selection is made. Selecting outside any of the menus appearing in the interaction cancels the interaction and returns NIL.

KEYWORD MENU

A keyword menu appears as a single menu, but is really a data structure encompassing a number of menus which the operator can move between to find the item desired. Each piece is in two parts: the set of keywords for all the pieces, and the items in this piece (having this keyword). Each piece is a Chunk Menu, so that if a particular keyword has many items, its piece is itself broken into pieces. The items in a keyword menu are computed from a list of objects, the things which presumably are being selected among. A function is provided for computing the keywords of each object, and another for computing an item from an object.

(KEYWORD.MENU.CREATE *OBJECTS KEYWORDFN PROPERTIES ITEMFN*) [Function]

Creates and returns a Keyword Menu, a data structure of pieces associated with the keywords of the *OBJECTS* as determined by *KEYWORDFN*. Each piece of the Keyword Menu is a Chunk Menu and contains the items for the objects with the associated keyword. The item is computed from the object by applying the *ITEMFN* to that object; *ITEMFN* should return an item appropriate for standard menus. The menus are determined by the values on the PList *PROPERTIES*. The properties understood are: CHUNK.COUNT (see Chunk Menus), TITLE, CENTERFLG, MENUFONT, ITEMWIDTH, ITEMHEIGHT, MENUBORDERSIZE, and MENULAYOUTSIZE. The actual menus are created only when needed.

(KEYWORD.MENU.INVOKE *KEYWORD.MENU POSITION*) [Function]

Carries out the interaction with the user to select an item from a Keyword Menu. If *POSITION* is non-NIL, the menu will appear at that position, otherwise it will appear under the mouse. All interactions are as with Chunk Menus, which is just that for standard menus for small numbers of items. Returns the value produced by the items when a selection is made. Selecting outside any of the menus appearing in the interaction cancels the interaction and returns NIL.