NAME

Curses::Widgets::Menu − Menu Widgets

MODULE VERSION

`$Id:` Menu.pm,v 1.103 2002/11/14 01:26:34 corliss Exp corliss $

SYNOPSIS

```
use Curses::Widgets::Menu;

$menu = Curses::Widgets::Menu−>new({
  COLUMNS     => 10,
  INPUTFUNC   => \&scankey,
  FOREGROUND  => undef,
  BACKGROUND  => 'black',
  FOCUSSWITCH => "\t",
  X           => 1,
  Y           => 1,
  MENUS       => {
    MENUORDER => [qw(File)],
    File => {
      ITEMORDER => [qw(Save Quit)],
      Save      => \&Save,
      Quit      => \&Quit,
    },
  CURSORPOS   => 'File',
  BORDER      => 1,
  });

$menu−>draw($mwh, 1);
$menu−>execute;

See the Curses::Widgets pod for other methods.
```

REQUIREMENTS

Curses
Curses::Widgets
Curses::Widgets::ListBox

DESCRIPTION

Curses::Widgets::Menu provides simplified OO access to menus. Each item in a menu can be tied to a subroutine reference which is called when selected.

METHODS

new (inherited from Curses::Widgets)

```
$menu = Curses::Widgets::Menu−>new({
  INPUTFUNC   => \&scankey,
  FOREGROUND  => undef,
  BACKGROUND  => 'black',
  FOCUSSWITCH => "\t",
  MENUS       => {
    MENUORDER => [qw(File)],
    File      => {
      ITEMORDER => [qw(Save Quit)],
      Save      => \&Save,
      Quit      => \&Quit,
    },
  CURSORPOS   => 'File',
  BORDER      => 1,
  });
```

The new method instantiates a new Menu object. The only mandatory key/value pairs in the configuration hash are **X** and **Y**. All others have the following defaults:

```
Key           Default  Description
=============================================================
INPUTFUNC   \&scankey  Function to use to scan for keystrokes
FOREGROUND      undef  Default foreground colour
BACKGROUND    'black'  Default background colour
FOCUSSWITCH      "\t"  Characters which signify end of input
MENUS              {}  Menu structure
CURSORPOS          ''  Current position of the cursor
BORDER              0  Avoid window borders
```

The **MENUS** option is a hash of hashes, with each hash a separate menu, and the constituent hashes being a Entry/Function pairs. Each hash requires a special key/value pair that determines the order of the items when displayed. Each item is separated by two spaces.

draw

```
$menu−>draw($mwh, 1);
```

The draw method renders the menu in its current state. This requires a valid handle to a curses window in which it will render itself. The optional second argument, if true, will cause the selection cursor to be rendered as well.

popup

```
$menu−>popup;
```

This method causes the menu to be displayed. Since, theoretically, the menu should never be seen unless it’s being actively used, we will always assume that we need to draw a cursor on the list as well.

execute

```
$menu−>execute;
```

This method acts like the standard Curses::Widgets method of the same name, with the exception being that selection of any menu item will also cause it to exit (having already called the associated item subroutine).

HISTORY

2002/10/17 — First implementation

AUTHOR/COPYRIGHT

(c) 2001 Arthur Corliss (corliss AT digitalmages DOT com)