cursesmenu Documentation

Release 0.4.1

Author

February 17, 2016

Contents

Contents:

CHAPTER 1

Installation

Windows users should visit here and download the curses build appropriate to your machine and version of Python.

Everyone should run:

```
pip install curses-menu
```

CHAPTER 2

Usage

First things first, import the package:

```
import cursesmenu
```

Or just import what you need:

```
from cursesmenu import CursesMenu

from cursesmenu.items import FunctionItem, SubmenuItem, CommandItem
```

Then create a menu:

```
menu = CursesMenu("This is a menu!", "It has a subtitle too!")
```

Create menu items for each choice you need:

```
command_item = CommandItem("Run a console command", "touch hello.txt")

function_item = FunctionItem("Call a function", input, ["Enter some input"])
```

To add other menus as submenus, use a *SubmenuItem*, setting the menu property in the constructor so the submenu's parent is set properly:

```
submenu = CursesMenu("This is the submenu")

submenu_item = SubmenuItem("Show a submenu", submenu, menu=menu)
```

Add the items to the menu:

```
menu.append_item(command_item)

menu.append_item(function_item)

menu.append_item(submenu_item)
```

Then start the menu:

```
menu.start()
```

After that, the menu will spawn its own thread and go about its business. If you want to wait on the user to finish with the menu before continuing, call:

```
menu.join()
```

To combine these two and simply show a menu and immediately wait for the user to exit the menu, call:

```
menu.show()
```

2.1 Getting a selection

If you have a list of strings, and you want to allow the user to select one, you can use a *SelectionMenu*:

```
from cursesmenu import SelectionMenu

a_list = ["red", "blue", "green"]

selection = SelectionMenu.get_selection(a_list)
```

Which is equivalent to:

```
from cursesmenu import SelectionMenu

a_list=["red", "blue", "green"]

menu = SelectionMenu(a_list,"Select an option")

menu.show()

menu.join()

selection = menu.selected_option
```

CHAPTER 3

API Reference

3.1 CursesMenu — Standard menu class

class cursesmenu.**CursesMenu**(*title=None*, *subtitle=None*, *show_exit_option=True*)

A class that displays a menu and allows the user to select an option

Variables

- **cls.currently_active_menu** (CursesMenu) – Class variable that holds the currently active menu or None if no menu is currently active (E.G. when switching between menus)
- **title** (*str*) – The title of the menu
- **subtitle** (*str*) – The subtitle of the menu
- **show_exit_option** (*bool*) – Whether this menu should show an exit item by default. Can be overridden when the menu is started
- **items** (list[*MenuItem*]) – The list of MenuItems that the menu will display
- **parent** (CursesMenu) – The parent of this menu
- **previous_active_menu** (CursesMenu) – the previously active menu to be restored into the class's currently active menu
- **current_option** (*int*) – The currently highlighted menu option
- **current_item** (MenuItem) – The item corresponding to the menu option that is currently highlighted
- **selected_option** (*int*) – The option that the user has most recently selected
- **selected_item** (MenuItem) – The item in items that the user most recently selected
- **returned_value** – The value returned by the most recently selected item
- **screen** – the curses window associated with this menu
- **normal** – the normal text color pair for this menu
- **highlight** – the highlight color pair associated with this window

start(*show_exit_option=None*)

Start the menu in a new thread and allow the user to interact with it. The thread is a daemon, so *join()* should be called if there's a possibility that the main thread will exit before the menu is done

Parameters **show_exit_option** (*bool*) – Whether the exit item should be shown, defaults to the value set in the constructor

join(*timeout=None*)
Should be called at some point after *start()* to block until the menu exits. :param Number timeout: How long to wait before timing out

show(*show_exit_option=None*)
Calls start and then immediately joins.

Parameters **show_exit_option** (*bool*) – Whether the exit item should be shown, defaults to the value set in the constructor

append_item(*item*)
Add an item to the end of the menu before the exit item

Parameters **item** (MenuItem) – The item to be added

add_exit()
Add the exit item if necessary. Used to make sure there aren't multiple exit items

Returns True if item needed to be added, False otherwise

Return type bool

remove_exit()
Remove the exit item if necessary. Used to make sure we only remove the exit item, not something else

Returns True if item needed to be removed, False otherwise

Return type bool

get_input()
Can be overridden to change the input method. Called in *process_user_input()*

Returns the ordinal value of a single character

Return type int

process_user_input()
Gets the next single character and decides what to do with it

draw()
Redraws the menu and refreshes the screen. Should be called whenever something changes that needs to be redrawn.

go_to(*option*)
Go to the option entered by the user as a number

Parameters **option** (*int*) – the option to go to

go_up()
Go up one, wrap to end if necessary

go_down()
Go down one, wrap to beginning if necessary

select()
Select the current item and run it

exit()
Signal the menu to exit, then block until it's done cleaning up

is_alive()

Returns True if the thread is still alive, False otherwise

wait_for_start(*timeout=None*)
Block until the menu is started

Parameters **timeout** – How long to wait before timing out

Returns False if timeout is given and operation times out, True otherwise. None before Python 2.7

pause()
Temporarily pause the menu until resume is called

resume()
Sets the currently active menu to this one and resumes it

is_running()

Returns True if the menu is started and hasn't been paused

3.2 SelectionMenu — Quickly get a selection

Bases: *cursesmenu.CursesMenu*

class cursesmenu.**SelectionMenu**(*strings*, *title=None*, *subtitle=None*, *show_exit_option=True*)
A menu that simplifies item creation, just give it a list of strings and it builds the menu for you

Variables **strings** (*list[str]*) – The list of strings this menu should be built from

classmethod **get_selection**(*strings*, *title='Select an option'*, *subtitle=None*, *exit_option=True*, *_menu=None*)
Single-method way of getting a selection out of a list of strings

Parameters

- **strings** (*list[str]*) – the list of string used to build the menu
- **_menu** (*list*) – should probably only be used for testing, pass in a list and the created menu used internally by the method will be appended to it

3.3 Items

3.3.1 CommandItem

Bases: *cursesmenu.items.ExternalItem*

class cursesmenu.items.**CommandItem**(*text*, *command*, *arguments=None*, *menu=None*, *should_exit=False*)
A menu item to execute a console command

Variables

- **command** (*str*) – The console command to be executed
- **arguments** (*list[str]*) – An optional list of string arguments to be passed to the command
- **exit_status** (*int*) – the exit status of the command, None if it hasn't been run yet

action()
This class overrides this method

get_return()

Returns the exit status of the command

Return type int

3.3.2 ExitItem

Bases: *cursesmenu.items.MenuItem*

class cursesmenu.items.**ExitItem**(*text='Exit'*, *menu=None*)
Used to exit the current menu. Handled by *cursesmenu.CursesMenu*

show(*index*)
This class overrides this method

3.3.3 ExternalItem

Bases: *cursesmenu.items.MenuItem*

class cursesmenu.items.**ExternalItem**(*text*, *menu=None*, *should_exit=False*)
A base class for items that need to do stuff on the console outside of curses mode. Sets the terminal back to standard mode until the action is done. Should probably be subclassed.

clean_up()
This class overrides this method

set_up()
This class overrides this method

3.3.4 FunctionItem

Bases: *cursesmenu.items.ExternalItem*

class cursesmenu.items.**FunctionItem**(*text*, *function*, *args=None*, *kwargs=None*, *menu=None*, *should_exit=False*)
A menu item to call a Python function

Variables

- **function** – The function to be called
- **args** (*list*) – An optional list of arguments to be passed to the function
- **kwargs** (*dict*) – An optional dictionary of keyword arguments to be passed to the function
- **return_value** – the value returned by the function, None if it hasn't been called yet.

action()
This class overrides this method

get_return()

Returns The return value from the function call

3.3.5 MenuItem

class cursesmenu.items.**MenuItem**(*text*, *menu=None*, *should_exit=False*)
A generic menu item

Variables

- **text** (*str*) – The text shown for this menu item
- **menu** (CursesMenu) – The menu to which this item belongs
- **should_exit** (*bool*) – Whether the menu should exit once this item's action is done

action()
Override to carry out the main action for this item.

clean_up()
Override to add any cleanup actions necessary for the item

get_return()
Override to change what the item returns. Otherwise just returns the same value the last selected item did.

set_up()
Override to add any setup actions necessary for the item

show(*index*)
How this item should be displayed in the menu. Can be overridden, but should keep the same signature.

Default is:

1 - Item 1

2 - Another Item

Parameters **index** (*int*) – The index of the item in the items list of the menu

Returns The representation of the item to be shown in a menu

Return type str

3.3.6 SelectionItem

Bases: *cursesmenu.items.MenuItem*

class cursesmenu.items.**SelectionItem**(*text*, *index*, *menu=None*)
The item type used in *cursesmenu.SelectionMenu*

Variables **index** (*int*) – The index of this item in the list used to initialize the *cursesmenu.SelectionMenu*

get_return()

Returns The index of this item in the list of strings

Return type int

3.3.7 SubmenuItem

Bases: *cursesmenu.items.MenuItem*

class cursesmenu.items.**SubmenuItem**(*text*, *submenu*, *menu=None*, *should_exit=False*)
A menu item to open a submenu

Variables **self.submenu** (CursesMenu) – The submenu to be opened when this item is selected

action()
This class overrides this method

clean_up()
This class overrides this method

get_return()

Returns The returned value in the submenu

set_menu(*menu*)
Sets the menu of this item. Should be used instead of directly accessing the menu attribute for this class.

Parameters **menu** (CursesMenu) – the menu

set_up()
This class overrides this method

3.4 Functions

cursesmenu.**clear_terminal**()
Call the platform specific function to clear the terminal: cls on windows, reset otherwise

cursesmenu.old_curses_menu.**parse_old_menu**(*menu_data*)
Take an old-style menuData dictionary and return a CursesMenu

Parameters **menu_data** (*dict*) –

Returns A new CursesMenu

Return type *CursesMenu*

CHAPTER 4

Indices and tables

- genindex
- modindex
- search