**Installation:**

The first thing to do is if you have a Windows computer is visit <http://www.lfd.uci.edu/~gohlke/pythonlibs/#curses> and download the proper version of curses depending on which python you have running on your computer. Then you want to go to the directory of the downloaded version in the command prompt and run “pip install [name of version chosen]”. This should now allow Window users to have the curses library on their computers. For Linux and Mac users

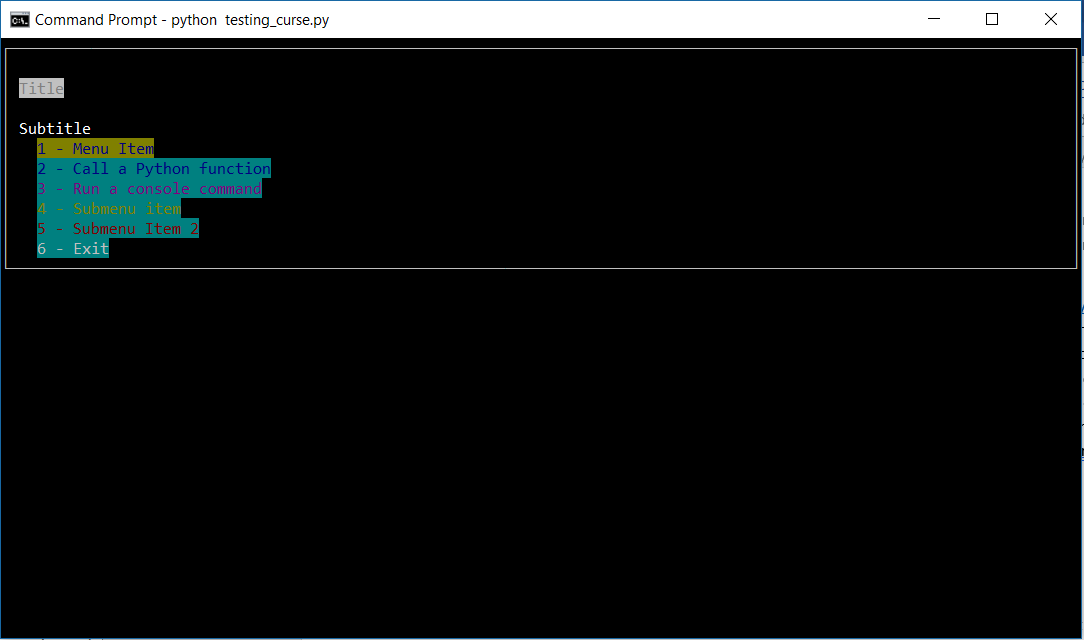
Now, go to Github and type in this url: <https://github.com/Cpizarro91/curses-menu/tree/dev>, this is the location of the updated curses-menu with all the improvements added. Depending on what IDE is being used, cloning this project should be easy. First, click on the “Clone or download” button to copy the necessary URL. If using Pycharm, create a new project then go to the VCS tab and scroll down to “Git”, select “clone…” and paste the copied URL here besure to clone the dev branch as that is where all the new updates are. If the user is using a Mac or Linux machine the install is simpler. Just go to the command line and type in “git clone [insert URL here]” and that will install the repository straight to the user’s computer and once again be sure to check that you are in the dev branch.

To check if a person is in the proper branch, from a Linux/Mac computer go to the command line and type in “git branch” and it might say master. If so, type the command “git checkout dev”, this will allow the user to switch to the proper branch to try. In Pycharm, go to the bottom of the screen and select “terminal”, then follow the same commands as the Linux/Mac users.

Now the program has been fully installed and is ready for use.

**Running:**

There is already a file in the program that runs the commands and allows a user to see a compilation of a curses menu. Go to the folder that holds curses menu and type in “python testing-curse.py”, what should appear is a curses menu that looks like this:



If the user would prefer to initialize the menu themselves the steps to do so are here:

# Import the necessary packages

from cursesmenu import \*

from cursesmenu.items import \*

# Create the menu

menu = CursesMenu("Title", "Subtitle")

# Create some items

# MenuItem is the base class for all items, it doesn't do anything when selected

menu\_item = MenuItem("RED", "Menu Item" )

# A FunctionItem runs a Python function when selected

function\_item = FunctionItem("BLUE","Call a Python function", input, ["Enter an input"])

# A CommandItem runs a console command

command\_item = CommandItem("MAGENTA","Run a console command", "touch hello.txt")

# A SelectionMenu constructs a menu from a list of strings

selection\_menu = SelectionMenu(["item1", "item2", "item3"])

# A SubmenuItem lets you add a menu (the selection\_menu above, for example)

# as a submenu of another menu

submenu\_item = SubmenuItem("YELLOW", "Submenu item", selection\_menu, menu)

# A MultiMenu constructs a menu from a list of strings

multi\_menu = MultiMenu(["Item 1", "Item 2", "Item 3"])

#for filter you have to put in [ ]

# A SubmenuItem used to create a menu within a menu using MultiMenu

submenu\_item2 = SubmenuItem("GREEN", "Submenu Item 2", multi\_menu, menu)

# Once we're done creating them, we just add the items to the menu

menu.append\_item(menu\_item)

menu.append\_item(function\_item)

menu.append\_item(command\_item)

menu.append\_item(submenu\_item)

menu.append\_item(submenu\_item2)

# Finally, we call show to show the menu and allow the user to interact

menu.show()

Each of these commands allow the user to create a menu with the specifications that they choose. For instance if the user wants a simple menu with nothing but the backscreen the user would input the first three lines above and the last one:

from cursesmenu import \*

from cursesmenu.items import \*

menu = CursesMenu("Title", "Subtitle")

menu.show()

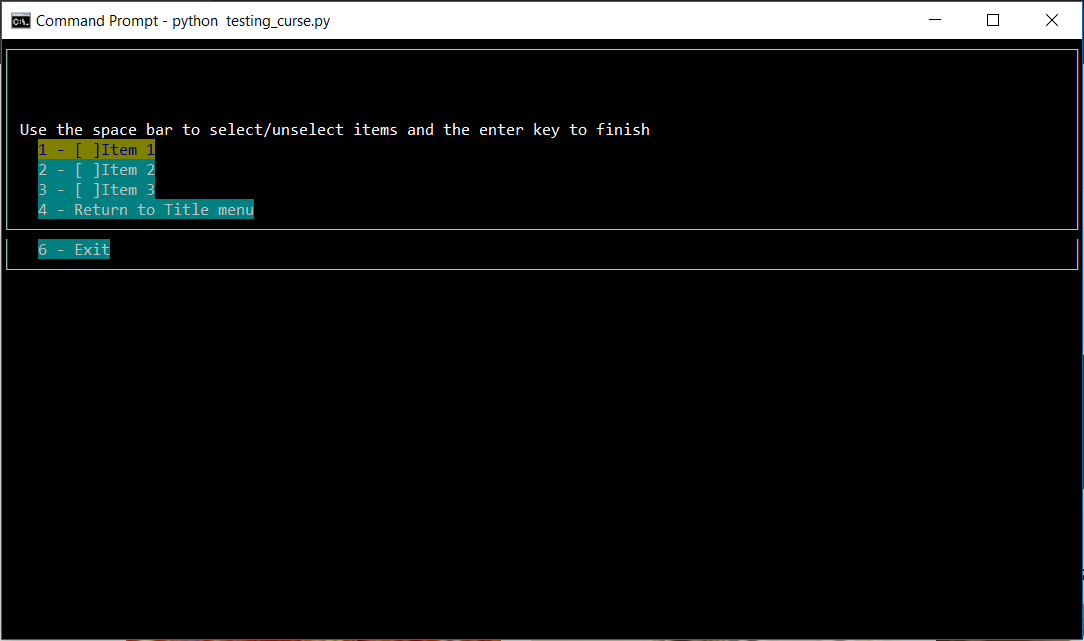
**Testing:**

Testing the program will require the user to install behave on their computer and make sure that they run it with Python 3.5, because all other versions of Python do not work well with the behave functionality.

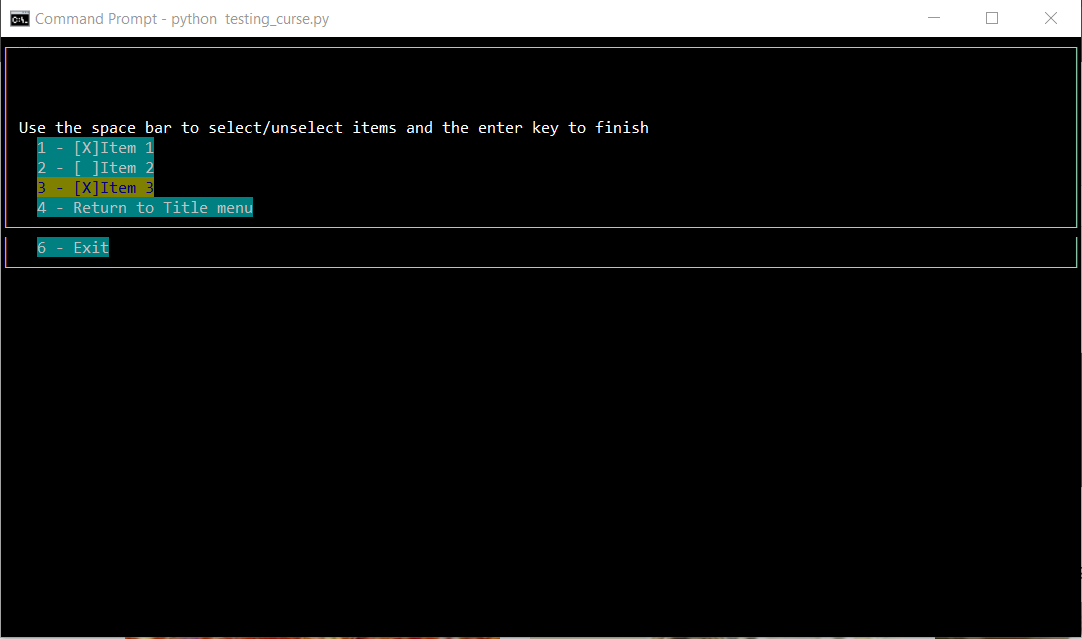
In the “features” file there are tests for each newly developed function of the curses menu. Which includes colors, multi-selection, and other updated or built functions. For testing the multi-selection function, the file will test the different implementations of creating a menu and submenus then will check if the action does select from a list of items or errors out.

**What you should be able to see:**

The menu was originally designed to only accept single selection of items, so one at a time. This was limited and very unuseful. When using the multi-select menu type the menu should look at first completely clean list with open brackets next to each item like so,



Then when a few items are chosen it should look like this,



Simple right?

**Current & Future Ideals:**

Right now, the program is able to allow the user to select an item and unselect an item, and only that since the issue documented was not very specific to what should have been done. That means right now the program only allows the user to choose what item they want in the submenu but no other actions occur. In the future, this function will be able to list a string of the chose items and allow the user to delete, add, or move that list to another menu.