Basic Curses with Python

Python 2.x 2.0–2.7 , Python 3.x 3.0–3.6

Examples

Basic Invocation Example

```
import curses
import traceback

try:
    # -- Initialize --
    stdscr = curses.initscr()   # initialize curses screen
    curses.noecho()             # turn off auto echoing of keypress on to screen
    curses.cbreak()             # enter break mode where pressing Enter key
                                #  after keystroke is not required for it to register
    stdscr.keypad(1)            # enable special Key values such as curses.KEY_LEFT etc

    # -- Perform an action with Screen --
    stdscr.border(0)
    stdscr.addstr(5, 5, 'Hello from Curses!', curses.A_BOLD)
    stdscr.addstr(6, 5, 'Press q to close this screen', curses.A_NORMAL)

    while True:
        # stay in this loop till the user presses 'q'
        ch = stdscr.getch()
        if ch == ord('q'):
    break

    # -- End of user code --

except:
    traceback.print_exc()     # print trace back log of the error

finally:
    # --- Cleanup on exit ---
    stdscr.keypad(0)
    curses.echo()
    curses.nocbreak()
    curses.endwin()
```

The wrapper() helper function.

While the basic invocation above is easy enough, the curses package provides the `wrapper(func, ...)` helper function. The example below contains the equivalent of above:

```
main(scr, *args):
    # -- Perform an action with Screen --
    scr.border(0)
    scr.addstr(5, 5, 'Hello from Curses!', curses.A_BOLD)
    scr.addstr(6, 5, 'Press q to close this screen', curses.A_NORMAL)

    while True:
        # stay in this loop till the user presses 'q'
        ch = scr.getch()
        if ch == ord('q'):

curses.wrapper(main)
```

Here, wrapper will initialize curses, create `stdscr` , a WindowObject and pass both stdscr, and any further arguments to `func` . When `func` returns, `wrapper` will restore the terminal before the program exits.

Syntax

Parameters

Remarks

Curses is a basic terminal (or character display) handling module from Python. This can be used to create Terminal based User interfaces or TUIs.

This is a python port of a more popular C library 'ncurses'