

Forum:[Hack Forums](#)

Thread: [Console Kit | Interactive Menu, Tables, Input Validation](#)

Odus - December 22, 2015, 4:17 pm

After seeing [N3w8i3H4ck3r's console menu](#), I was going to post some suggestions for improvement but got carried away and ended up writing this handy tool kit for making the console more user friendly with interactive menus, easy validation and tables.

Demonstration: [\(Click to View\)](#)

[img]http://orig05.deviantart.net/4b83/f/2015/365/4/0/2015_12_31_16_27_59_by_oduslabs-d9m6mx5.gif[/img]

Interactive Menu

Looks a bit more pleasant than prompting the user to enter a choice corresponding to a printed menu.

All you have to do is set your options and console highlight color. Originally the options were going to be passed through a ParamArray in the constructor but I ended up making them properties so it wouldn't need to be re-instantiated if you had branches or multiple menus.

There's also some recursion in there which lets the function itself handle (what appears to be) the scrolling... hnnngggg

Usage:

```
Code var menu = new Menu()
{
    Options = new string[] { "trvp", "is", "beautiful"},
    HighlightColor = ConsoleColor.Red
};
```

```
var selection = menu.AwaitInput();
```

Data Table

For easy data formatting, a table builder is also included specifically for custom data models. Our test case model looks like this:

```
Code public struct User
{
    public string Name { get; set; }
    public string Surname { get; set; }

    public int Age { get; set; }
}
```

It fetches all public properties' names and values used to populate the table headers & rows. There are also 2 args passed in the constructor for dimensions: width and indent. The first sets the span across the console, the second is for spacing between the console window border and the bounds of the grid - purely aesthetic.

Usage:

```
Code var users = new List<User> { }; //assume populated
```

```
var table = new Table(50, 2);
```

```
table.BuildTable(users);
```

Validator

Writing a bunch of while loops and constant checking can get very boring, so it's wrapped up nicely here for you. To keep things neat, a colon and space is automatically appended to your prompt. You can get input, which simply prompts for input and attempts to convert it to the type you specified. If it can't, it allowed an error message to show and reattempts input.

Alternatively, you can validate, which works similar but supports a lambda expression to check for validation. This could be handy for anything, such as string length, int range, etc. As long as your expression returns a bool, you're good to go.

Usage:

```
Code var age = Validator.ValidateInput<int>("Enter your age", "Input must be a number", (i) => i > 18);
```

Downloads:

<https://github.com/oduslabs/ConsoleKit> (ConsoleKit.cs)

Unknown - December 22, 2015, 6:44 pm

10/10 would use. I need to make a Console App now o.o

N3w8i3H4ck3r - December 23, 2015, 8:37 am

Well that is much easier to do, thank u for taking a look at my code and linking my thread!

debug - December 23, 2015, 10:23 am

Ahh, recursion. Recursion was almost the death of me when I took AP Comp sci last year. Anyways, looks very nice. Good job odoose.

SIM-1676753 - December 23, 2015, 10:50 am

Cute release. Looking over it now.

poopGod - December 23, 2015, 8:01 pm

I give it a perfect 5/7, Trvp is beautiful. Also, nice release.

Moen - December 23, 2015, 10:52 pm

Posting here so i have a higher possibility to find this when i actually need it.

Odus - December 24, 2015, 1:48 pm

Thank you guys. Enjoy!

(Dec 23, 2015 09:01 PM)poopGod Wrote: [CITE](#) I give it a perfect 5/7, Trvp is beautiful. Also, nice release.

I see. What would give it that 8/8 r8?

Programming Enthousiast - December 24, 2015, 4:22 pm

I do not know Mr. Trvp but considering his user title and this post I can assume he is a majestic human being.

Jokes aside this seems nice, Odus. Maybe add like a progressbar property to it? Just have ? character appended to a string which is displayed until it reaches a certain length (like a 100 or

whatever the user specifies). And have it so people can set it to a certain value or something. Seems like a cool idea.

GhostCode - December 25, 2015, 9:36 am

Little suggestion @ Odus :

However about adding the options for a custom amount of submenus for every menu entry?
Something like ListViewItem & ListViewSubItems?

Odus - December 25, 2015, 9:59 am

(Dec 25, 2015 10:36 AM)GhostCode Wrote: [\[CITE\]](#) Little suggestion @ Odus :

However about adding the options for a custom amount of submenus for every menu entry?
Something like ListViewItem & ListViewSubItems?

That was actually another reason behind including the menu options as a property because it caters only to the options and the result. Accounting for multi dimensional arrays (i.e. the menu tree) would add complexity to the class. Everything from going through the branches to returning the result.

nuk - December 25, 2015, 6:32 pm

This is EPIC! Def. gonna come in handy ;) Thanks for writing this up ;)

Milopeach - December 26, 2015, 4:42 am

This is very nice, indeed! Great job man. I could see myself failing miserably at making something like this, I don't know why.

Aristocles - December 28, 2015, 1:40 pm

can't get it to work : Console.WriteLine(\$"{i + 1}.\\t{Options[i]}"); vs does not recognize the \$ symbole, .net 4.6 vs community

Odus - December 28, 2015, 1:45 pm

(Dec 28, 2015 02:40 PM).NETGuard Wrote: [\[CITE\]](#) can't get it to work : Console.WriteLine(\$"{i + 1}.\\t{Options[i]}"); vs does not recognize the \$ symbole, .net 4.6 vs community

You must not be using C# 6.0 is my guess. Edited back to the old string interpolation, try again.

Aristocles - December 28, 2015, 1:51 pm

thanks Odus; How to upgrade to c# 6.0 ?

Odus - December 28, 2015, 2:02 pm

(Dec 28, 2015 02:51 PM).NETGuard Wrote: [\[CITE\]](#) thanks Odus; How to upgrade to c# 6.0 ?

Go to your project properties -> build -> advanced -> language version. It should be an option there.

Odus - December 31, 2015, 9:10 am

Since a console project came up, I rewrote this into a full kit with a bunch of stuff so here ya go. ^^

Working on the VB version now.

SaketSaket - January 3, 2016, 2:49 am

This is awesome.... & very useful... saving for future use... ;)

SearchEngineer - January 3, 2016, 3:24 am

Do you know if this can be done in autoit?

So instead of a GUI with a "Launch" button, I could use Console line and press 1 for start etc?

Marwix - January 3, 2016, 4:09 am

Just did a test myself, and I have to tell.. it's pretty nice.

I really like the way you navigate through the menu on the console.

Good job!

Odus - January 3, 2016, 8:25 am

(Jan 3, 2016 04:24 AM)SearchEngineer Wrote: [\[CITE\]](#) Do you know if this can be done in autoit?
So instead of a GUI with a "Launch" button, I could use Console line and press 1 for start etc?
I don't see why not, can't imagine the algorithm changing. The VB version is already pretty pseudo to convert but I have no experience using AutoIT.

(Jan 3, 2016 05:09 AM)Keiko Wrote: [\[CITE\]](#) Just did a test myself, and I have to tell.. it's pretty nice.

I really like the way you navigate through the menu on the console.

Good job!

Thank you. ^^

SIM-1676753 - March 31, 2016, 4:16 pm

(Mar 30, 2016 01:50 PM)xadamxk Wrote: [\[CITE\]](#) I WANT TO SEE SCREENSHOT EXAMPLES
ODUS!
Goddammit Adam compile the thing.

Odus - April 1, 2016, 12:45 am

(Mar 30, 2016 01:50 PM)xadamxk Wrote: [\[CITE\]](#) I WANT TO SEE SCREENSHOT EXAMPLES
ODUS!
The spoiler, dear boy. Click it ?

LaiLaiZhen - April 10, 2016, 6:00 am

This is a really awesome tutorial, im just finjshing my first year of college and we are learning C Sharp, this helped with the menu system for my graded unit project. Thanks.

Sleek. - April 11, 2016, 11:27 am

The github page for C# is down!

Odus - April 11, 2016, 1:19 pm

(Apr 11, 2016 11:27 AM)Strikecarl Wrote: [\[CITE\]](#) The github page for C# is down!
My bad, forgot to update the thread when I renamed the repo.

(Apr 11, 2016 01:09 PM)Electroalek Wrote: [\[CITE\]](#) Just stumbled upon this one, here is a new idea.
Create cursor support for console applications! Will be pretty unique.
You mean like point and click? Capturing clicks wouldn't be difficult but what are you thinking on mapping it to actions/handlers? Divide into regions?

Marshall Mathers - May 24, 2016, 3:25 pm

This is some good stuff, thanks bro.

Ankeras - September 7, 2016, 4:55 pm

Luv you man xd this will help a lot in class ! thx for sharing

0xDEAD10CC - September 8, 2016, 2:10 pm

Instead of using Convert.ChangeType() I would implement a custom TypeConverter class to allow

the user to convert from a string type to a custom user-defined type. Right now that is not possible, and although you could just construct it manually, the use of TypeConverter's would provide a bit more flexibility I think.

<https://msdn.microsoft.com/en-us/library/ayybcxe5.aspx>