Do Character User Interfaces have a future?

Asked 16 years, 1 month ago   Modified 5 years, 1 month ago

Viewed 3k times

9

We've got products built both with GUI and CHUI. Going forward, we're looking at redesigning a lot of our software and mainly taking the route of going all GUI. My question to the group is, do we need to account for keeping a CHUI around? What are the advantages of CHUI over GUI? Many times in the past people have said that CHUI is faster because you don't need a mouse. I argue that GUI can be just as fast with the right keyboard shortcuts, hotkeys and/or touch screens.

Is CHUI something we should no longer consider if hardware no longer provides a constraint?

Also to clarify, when I speak about CHUI I mean a CHaracter based User Interface, and I'm also mainly concerned with the effective presentation of data to an end user.

There have been some fantastic responses that have highlighted the importance of having a command line based interface for automation and scripting based tasks which I will certainly take to heart when we begin the design!

Share

Improve this question

Follow

asked Oct 31, 2008 at 14:16

Mat Nadrofsky
8,274 ● 8 ● 51 ● 73

I know GUI (Graphical User Interface) but what is GHUI?
– Patrick Desjardins Oct 31, 2008 at 14:30

---

1 CHUI stands for CHaracter User Interface. It's a common term in the "Progress 4GL" land which is one of the db/language technologies we use in-house.
– Mat Nadrofsky Oct 31, 2008 at 14:34

It's funny how most people answering think you are referring to a command line interface. – postfuturist Oct 31, 2008 at 15:23

---

15 Answers

Sorted by: Highest score (default) ⇕

▲

15

▼

🔖

You should poll your customers, not programmers. If your customers, who use your applications, want a CHUI, even if all your developers think it's a waste of time, you build it, because the customer is always right (except for when they're wrong).
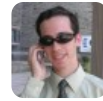
answered Oct 31, 2008 at 14:20

Elie
13.8k ● 25 ● 78 ● 128

2   "[...] the customer is always right (except for when they're wrong)." - Elie – Camilo Martin Mar 15, 2010 at 7:28

14

The primary benefits of a CHUI (that is something with forms and fields, not necessarily command line interfaces) is the keyboard for navigation and consistent layout. That is key.

If your GUI can be completely, and efficiently, keyboard navigated, then your CHUI user base should be happy. This is because in time, the users simply "type" their commands in to the system without "seeing the interface". They don't need to "discover" the interface, which is a primary feature of the GUI.

While CHUIs appear to be dinosaurs, they are still functional and usable. Most folks once they're trained (notably POS/Counter workers, but even back office scenarios like factory or warehouse floor, etc) have no problem using a CHUI.

But the key is the keyboard support so the user don't have to wait for the screen to catch up with them. Seeing a skilled operator with a mastery of the keyboard can make an application fly. You barely have a chance to see popup windows and what not.

Share   Improve this answer

Follow

1   I completely agree, you can fly through those screens with a few key presses. I've just never seen a GUI move that fast, even on very fast machines. – postfuturist Oct 31, 2008 at 15:21

You should absolutely still consider it. Most importantly, command line programs can be automated (and chained together in scripts) much more easily than GUIs (typically). I can't imagine working with a source control tool which didn't have a command line interface - although obviously having a GUI is useful too.

10

Now whether you need a command line version for your particular app is hard to say without knowing what your app does. Do you need automation and scripting? Might someone want to VPN in and run it from a very bad connection, and thus appreciate low bandwidth?

Note that MS certainly doesn't believe the command line is dead - or they wouldn't have created PowerShell.

Share   Improve this answer

Follow

a scriptable command line interface is not necessarily the same thing as a CHUI. You could easily package a gui app

that took command line args and ran without displaying an interface. – Nick Oct 31, 2008 at 14:36

Very well put. I certainly agree command line isn't dead. I was asking more in a sense of presenting information to an end-user. I'll edit the question to clarify slightly. You did raise an excellent point about having a command-line interface to our app that I'm going to bring up! – Mat Nadrofsky Oct 31, 2008 at 14:36

1 Ah, at first I misunderstood what CHUI meant, but there is all-mighty Jon Skeet to misunderstand the same thing and make me feel better. – Camilo Martin Mar 15, 2010 at 7:30

I agree with Eli that your customers should have final say, **but** if you can keep the meat of your program from being too interwoven with the GUI(or CHUI), then production cost to make both available should be minimal.

5

Share   Improve this answer

Follow

answered Oct 31, 2008 at 14:27

J.J.
4,892 ● 1 ● 26 ● 29

1 I wholeheartedly agree here. Gotta keep the layers separate! – Mat Nadrofsky Oct 31, 2008 at 14:33

5

If you write apps for unix and you need to handle users who telnet / ssh to your box then you will need command line interfaces.

I would say it depends on your target. Do you script your code from other apps? That would be a requirement to keep the interactive version (or some piece to avoid the GUI startup).

We usually do one or the other. But sometimes we have utils that have to be deployable through ftp and run ssh. Or we have tools that our users embed into their apps and don't want to expose a UI (data migration / conversion).

Share Improve this answer

Follow

answered Oct 31, 2008 at 14:28

Jason Short

5,314 ● 1 ● 31 ● 44

> Our particular application will likely have a strong linux based back end so telnet/ssh are important! – Mat Nadrofsky Oct 31, 2008 at 14:42

4

To this day, some of the most efficient user interfaces I've ever seen were plain old terminal-based character interfaces.

Anecdote: I was once part of a project to "modernize" a terminal application used by 500 customer service representatives. We published sexy GUI mockups and

everyone, including the users, were suitably impressed. We worked for six months on the application, and all the user acceptance testing seemed to indicate we had a winner.

But when the application was finally launched, it failed miserably. As it turns out, CSRs are measured for performance daily, right down to the average number of seconds per call handled. And no matter how hard they tried, they could not match the same level of efficiency in the GUI as they could in the terminal interface. They could get close with tabs and shortcuts, but not quite there.

Hard lessons learned. Modern programmers may abhor "dinosaurs", but do users really care about slick interfaces? Usually they just want to get their work done.

Share Improve this answer

Follow

3

When I first read this, my immediate thought was that this is probably one of those apps that's basically a series of forms, but displays inside a terminal. Often you see such dinosaurs running on cash registers. I also recall seeing such an app used to apply for a loan when I bought my car. This type of application doesn't seem to have a place in the modern world -- any system with even a tiny bit of processing power can handle a normal GUI nowadays. Unless you're trying to support really low-end legacy

customers, get rid of this user interface. A GUI with decent keyboard shortcuts (please, please, please put some thought into keyboard-only use of your GUI programs...) is going to be equally effective for the users coming from the old CHUI system and much friendlier to those used to a GUI, without having to have 2 versions of your app.

I don't see why everyone is bringing up command line apps. I think most people recognize that the command line isn't going away. It's far faster for many tasks than a GUI, largely because the programs tend to be non-interactive (and thus easily scriptable). As soon as your app becomes interactive (or, at least, doesn't have a param to make it non-interactive), running it from the command line is much less important. Even awesome programs like Vim that are terminal-based are transitioning to their graphical counterparts (gVim) because it gives you the best of both worlds.

Share   Improve this answer   answered Oct 31, 2008 at 14:44

Follow

rmeador

25.6k ● 18 ● 64 ● 104

Well put. This pretty much describes exactly where we're at. We've got a legacy based CHUI system that looks very similar to those old mainframe bank applications or cash register applications. I've got my own opinions, and I'm trying to see if I'm in-line with the masses or out to lunch. Thanks! – Mat Nadrofsky Oct 31, 2008 at 14:50

Um, about vim being awesome...*coughs and looks shifty* ...I disagree! – Paul Nathan Oct 31, 2008 at 15:12

2

Even GUI apps like Firefox can benefit from command line interfaces like Ubiquity. If there's a way to provide the command line from within the GUI then why not have the best of both worlds?

A lot of CAD programs have command line interfaces that show you what the GUI interaction you just performed equates to in the command line. That way you can learn the command line operations for the things that you do frequently and where the command line can be quicker to interact with whist still having the discoverability of the GUI interface.

See this youtube video demonstrating Rhino3D's command line

Share   Improve this answer

Follow

edited Oct 31, 2008 at 14:28

answered Oct 31, 2008 at 14:22

That video's not the best example. Anyone got a better one?
– Sam Hasler Oct 31, 2008 at 14:29

CHUI is faster in execution speed, not user interaction speed. I write embedded systems (as well as GUIs), so I'll always have a use for command line apps.

Share Improve this answer

Follow

answered Oct 31, 2008 at 14:19

Bill the Lizard
405k ● 211 ● 572 ● 889

Every study I have ever read showed that CHUI's are much faster for experienced users. GUI's are easier for new users and for applications that are only occasionally used. Also for a given screen size, you can display more information on a CHUI then a GUI. A good GUI can give you a quick over view at a glance.

Share Improve this answer

Follow

answered Oct 31, 2008 at 15:44

Jim C
4,981 ● 23 ● 25

I would watch a general statement on screen size in relation to CHUI. Don't forget some CHUI based interfaces have an 80 character screen width limit depending on the terminal they are running on. A CHUI that is run via a GUI based re-

sizable window on a PC can be. – Mat Nadrofsky Oct 31, 2008 at 15:58

In addition to the other benefits mentioned above, I've frequently found another reason to keep around an alternative UI--it keeps you and your interfaces honest. When an application is built with only one user interface, it becomes much easier to let design principles slide and for your business logic, etc. and your GUI to become an intertwined ball of spaghetti--despite best intentions. Regardless of the importance of your customers having a command-line interface, soon there might come a time when an alternative GUI (read: presentation layer) might be needed, and you'll want to be prepared. This might not be relevant to your requirements, but I think it's something good to keep in mind...

Share Improve this answer

Follow

answered Dec 28, 2008 at 17:11

reuben
3,370 ● 25 ● 28

One of the big issues that we encountered was multisession capability which is almost nonexistent with the GUI technologies I have seen. Our users were quick to point out that with the current character based interface they could have over a dozen Telnet based terminal sessions going at the same time on their PC screen which enabled them to multitask or task switch with high efficiency. They rated multitasking as the killer feature

which they benefitted from in our fast paced environment where interruptions are frequent. Being able to have concurrent access to multiple instances of a particular ERP application or multiple different ERP applications while always retaining session states was important to our user community.

Share   Improve this answer

Follow

answered Aug 7, 2009 at 2:38

Nigel Winterbottom

+1 - Well put and well said. A very real advantage of Character based stuff. I would say that you could build a GUI interface to behave the same way however. Though why you'd go to that effort for no other reason than to have it look like a window, I'm not entirely sure. ;) – Mat Nadrofsky Aug 7, 2009 at 18:42

1

I think the problem comes from design practices in GUI forms. We tend to place more objects on them especially with a vertical scroll bar and tab capabilities. This also makes loading slower. Going through CHUI menus with the keyboard is faster once you've memorized those sequences and holding the Ctrl key isn't required. There is something about the menu bar in Windows where the short-cut key descriptions are off to the right. The character based menus seemed easier to remember after awhile.

```
A) - This Menu
B) - That Menu
```

```
C) - Some other Menu
```

Or you could arrow through the choices and you just seemed to have some muscle memory where That Menu is the second choice.

Share   Improve this answer

Follow

As soon as you present some data, someone's going to want to query against it. You can integrate that with a gui, no problem. If you think some of your customers are going to want to script certain tasks. set it up. Anything to do with automation is better done from the command line(y harlo thar cron job!)

I love guis. I'm a mac user. But there is a time and a place for a CLI.

0

Share   Improve this answer

Follow

I was sysadmin at a university math department when the registration system went from a character based system using telnet, to a gui system on a PeopleSoft app.

0

The gals in the front office HATED the new system. Now part of this was the whole bit about old shoes being more

comfortable. But when I asked about it, Christine said that even after a week of doing several hundred registrations per day, the new system took several times as long to do anything. Lots of things only doable with a mouse. The old system could accept input as fast as they could type. Screen repaints were under a tenth of a second. New system had lots of 3/4 to 2 second pauses -- just long enough to be annoying, not long enough to do anything else.

Share   Improve this answer

Follow