Rules about disabling or hiding menu items [closed]

Asked 16 years, 3 months ago Modified 7 years, 3 months ago

Viewed 3k times

3

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed 2 years ago.

Improve this question

Have you ever been in a situation where a menu function you really really want to use but can't cause it's disabled or worse gone all together?

There is an argument for always leaving menus enabled and then display a message to a user explaining why a menu function can not be activated when they click on it. I think there is merit in this but maybe there is a cleverer way of addressing this issue.

I would be interested to hear what others think.

user-interface standards

Share

Improve this question

Follow

edited Sep 3, 2017 at 0:52

til

203 ● 2 ● 6

asked Sep 17, 2008 at 1:43

Crusty

141 ● 2 ● 6

A tooltip for each menu (disabled or not) may help the users understand its role (and why it was disabled). – IceCold Apr 28, 2015 at 13:07

8 Answers

Sorted by: Highest score (default)

▲

10

▼

If you're refering to Joel's post [Don't hide or disable menu items](), he [clarified in the StackOverflow podcast]() that he intended that there be information - not a dialog - telling you why a menu item wouldn't do anything:

> So, the use-case I was thinking of was, you had mentioned that in the Windows Media Player, you can play things faster when you're listening to podcasts and so forth, and it'll speed them up. And when I looked in there, that was disabled. And I couldn't figure out how to enable it. And obviously the help file is no help--not that anybody reads help files, but even if you did you

> couldn't find the answer to that. And that was kind of frustrating, and I'd rather have that menu item be enabled and have it just tell me "I'm not going to do this right now because of the following reason. I refuse to do this."

Share Improve this answer Follow

answered Sep 17, 2008 at 1:48

Josh

8,016 ● 5 ● 43 ● 63

▲

4

▼

As with most questions about usability, the answer is "it depends". It depends on the problem domain, the type of user, how critical the function is and so on. There is no single answer to your question.

I think the general consensus is, never ever totally remove items from a menu. Menus allow the user to freely discover what functions are available, but if those items are hidden or move around it does nothing to help the user. Also, moving them around makes it impossible to become proficient with the application since you have to constantly scan the menus for the item you want to select.

As for disabling versus enabling an item and displaying a dialog or message explaining why it's not something you can do, I generally prefer the former. However, if there's a function that a user can't reasonably be expected to intuit from the display, leaving it enabled is a good choice.

For example, if "Paste" is disabled it's reasonably obvious to most computer users that there's nothing to paste. However, if you have a "Frizzle the Bonfraz" menu item and the user may not know what a Bonfraz is or why they might want to enable it but can't, it's a good idea to leave it enabled at least for a while.

So again, it depends. If at all possible, do what you think is best and then ask your users.

Share Improve this answer Follow

answered Sep 17, 2008 at 1:52

Bryan Oakley
385k ● 53 ● 569 ● 727

What exactly do you mean with "leave it enabled at least for a while"? Would you first have the user click it, then show an explanation why that cannot be done then start disabling it whenever it isn't an option? I would find that frustrating. And it would give everyone the same experience/chance of learning in the case that many users use the same installation of that software. – masterxilo Jun 18, 2014 at 6:48

@masterxilo: yes, that is what I meant. I don't necessarily recommend ever doing such a thing, it all depends on the user, the app, and the context in which they use it.
– Bryan Oakley Jun 18, 2014 at 10:52

4

To generalize it a bit (perhaps incorrectly...), which of these situations would you prefer:

1. To find yourself on an island with no boat or bridge in site. Of course, you could talk to the villager in town and he would tell you the magical word to make a bridge appear...but you had no idea that magic existed.
2. You see that there is a bridge; however, when you get to it, there is a sign telling you that the bridge is not open to use.
3. You see that there is a bridge and celebrate! When you get to the end of the bridge, it tells you that the exit is not open. They must go back.

Maybe I am biased, but I don't believe that leaving the menu options enabled and allowing the user to click on it is the best of idea. That's just wasting someone's time. There is no way for them to distinguish that the item is available or not until they click on the item. (Scenario #3)

Hiding the item all together has its pros and cons. Completely hidden and you run the risk of the user never discovering all these features; however, at the same time, you are presented with the opportunity of making your application 'fun' and 'discoverable.' I've always thought the visibility of actions is more suited to items like toolbars. A good example of that is in when in some applications the picture toolbar pops up when you click on an image...and disappears when you click on text. In general, I would say that something like this is best if the overall experience of your application lends towards a "discovering" and "exploring" attitude from the user. (Scenario #1)

I would generally recommend disabling the items and providing a tooltip to the user informing them how to enable it (or even a link to Help?); however, this cannot be overdone. This must be done in moderation. (Scenario #2)

In general, when it's a context-related action (i.e. picture toolbar) that the user can easily discover, hide the items. If the user won't easily find it, have it disabled.

Share Improve this answer Follow

answered Sep 17, 2008 at 2:03

Swati
52.6k ● 4 ● 40 ● 54

What about situation 4: You see that there is a bridge, there is a sign telling you that the bridge is not open to use. However, you *can try* and when you get to the end, it tells you that the exit is not open *and why*. This means showing a "disabled but still clickable" menu item bound to an error message. – maaartinus Dec 18, 2013 at 20:58

1 I really like your anecdote with the island and the bridge. It's really hard to tell whether one would prefer 1 or 2. In situation 1, if you're not the kind of person to talk to the villager you might never know. In some cases, 2 might be similar to 3. If you are far away from the bridge, you might not be able to tell whether it's open or not, so you walk there only to have wasted your time. This is comparable to e.g. a nested submenu will all operations disabled. – masterxilo Jun 18, 2014 at 6:55

3

Make it disabled but have the tooltip explain why it's disabled

Share Improve this answer Follow

answered Sep 17, 2008 at 1:45

Ana Betts
74.6k ● 16 ● 143 ● 210

This had occurred to me too. Often tool tips are used to describe in a few words what the action does. Tool tips are not great for long sentences, or paragraphs of information. – Crusty Sep 17, 2008 at 1:52

1 Funny enough in many native windows controls tooltips are getting disabled as well :) – dr. evil May 30, 2010 at 21:40

I never said that Windows does it right, only that you have a chance to remedy our mistakes :) Disabling it lets the user quickly see that something has changed, so that once he *does* read the Tooltip, he knows and doesn't need to click the menu item every time only to find out, "Oops, it doesn't work" – Ana Betts May 31, 2010 at 23:06

0

It depends on the situation. If the menu item has applies in the current context but isn't available because of state, it should be disabled. If the context has changed so it no longer applies, it should be removed.

Share Improve this answer Follow

answered Sep 17, 2008 at 1:49

Joe Skora
14.9k ● 5 ● 38 ● 39

How exactly do you define "context" and "state"? For me context = current collection of state. But I guess with state you mean *large* phases the application can be in. For example with a 3d modeling application, 1. import & position, 2. editing, 3. 3d print preparation. But if you agree with that definition, the answer should switch it's use of *state* and *context*. – masterxilo Jun 18, 2014 at 7:01

0

I've never really understood this myself (I don't program GUIs). Why even have menu items hidden or disabled in the first place? It is non-intuitive for most users who are looking for a particular menu option to find it disabled, or not even present.

Tooltips are also non-intuitive. If I'm moving my mouse across menu items, I'm not going to pause long enough to get a tooltip explanation. I'm more likely to become frustrated that something I expected to be accessible through the menu isn't there, or is disabled.

That said, I actually don't use GUI menus very often. I find the options available are often not useful, or are accessible in some more intuitive way, such as common keyboard shortcuts.

Share Improve this answer Follow

answered Sep 17, 2008 at 1:51

jtimberman

8,258 ● 2 ● 45 ● 37

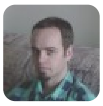
0

You can display the 'reason' in the status bar. Or even better, use a text that describes the action and contains information when such action is possible. For example, for 'Copy' menu item, the text in status bar would be: Copy the selected text. Note the 'selected' part, which tells the user that he needs to select the text to enable the menu item.

Another example in a tool I'm making, we have 'Drop database' menu item, but this action is only possible when you're connected to it. So, the text in status bar goes something like: 'Drop the database (only when connected)'.

Share Improve this answer Follow

answered Sep 17, 2008 at 2:56



Milan Babuškov
61k ● 49 ● 130 ● 180

Good point about making the text more explanatory. To apply it to your second example, why not have it say "Drop connected database", then the user doesn't need to look to the status bar. – masterxilo Jun 18, 2014 at 6:58

0

I've always believed that you should hide as much as you can. (Your application shouldn't be any more complex than what the user can/should do.)

If you display a menu option that a user shouldn't be using, they may click on it, but think your application is broken because nothing happens.

That's what I think at least...

Share Improve this answer Follow

edited Sep 18, 2008 at 19:25

answered Sep 17, 2008 at 1:47

Mikey
2,837 ● 2 ● 18 ● 17

I normally only hide things when security privileges prevent a user from performing an action. – Crusty Sep 17, 2008 at 2:02

That's basically what I meant. I don't think you should show menu options to users that don't have the correct permissions to perform that cooresponding action. – Mikey Apr 23, 2009 at 12:47