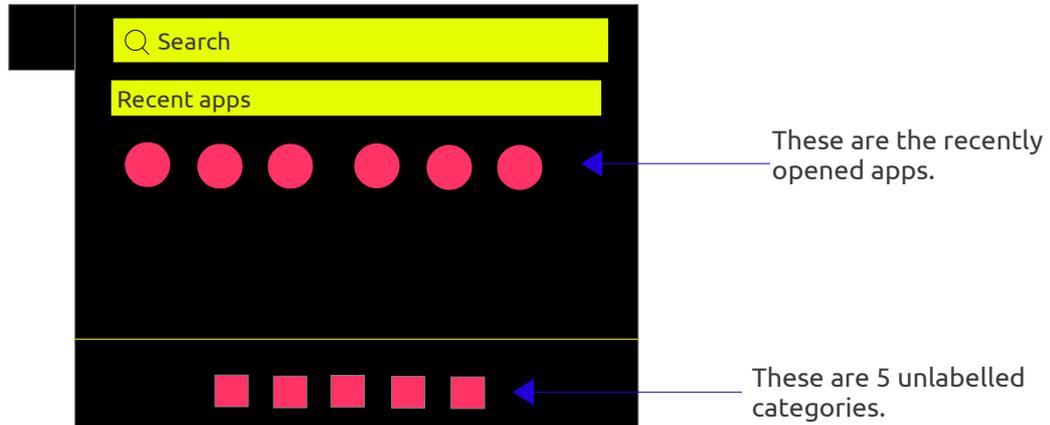


Suggestions for the Unity interface in Ubuntu.

Ideas for tweaking Unity in many areas. Diagrams are not to scale. May 2012.
Tested in the Evince PDF viewer.

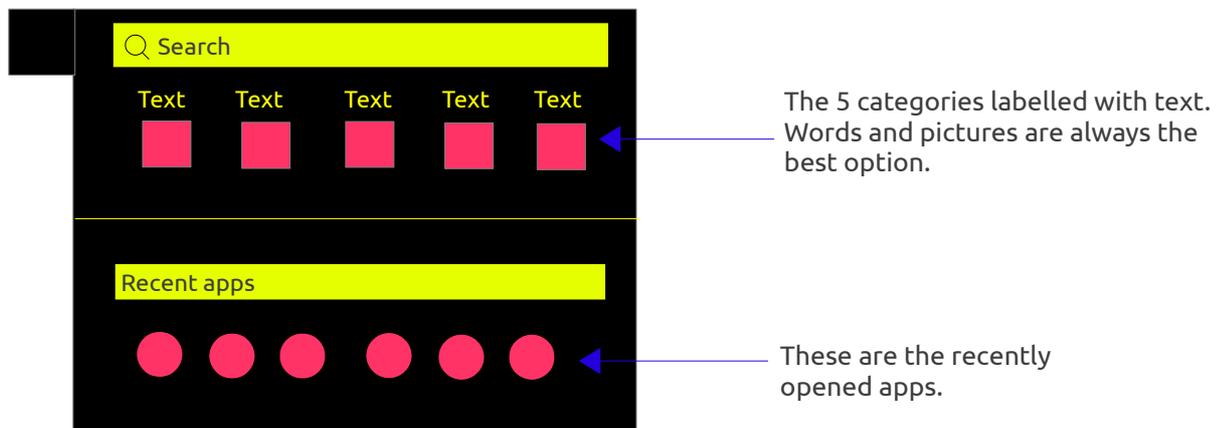
Dash Home

Current:



Improvements:

In order to keep the Unity interface consistent throughout, the “top to bottom” layout can be implemented. Unlike in the current layout – which requires the user to look at the bottom of the lens and then look up again, moving the categories to the top allows the user to start looking at the top and work their way down.



Scroll bars

The current scroll bars are very thin and not noticeable in the background. Not every user likes the “edge scrolling” and/or “tap to click” enabled on notebook touchpads, and those users' who have turned off “edge scrolling” will be using the scroll bars directly. Fortunately it's possible to make the scroll bars work well and save screen space, as explained on the next page.

If screen space is limited e.g. on netbooks, it is possible to have the scrollbar thin or wider when required, as illustrated below.

Diagram 1 shows a lens with a visible but thin scrollbar. Notice that the up/down arrows clearly shows the user that the lens has a scrollbar. By using *permanently* visible up/down arrows (or left/right arrows), users can instantly recognise scroll bars and this could be implemented on every scroll bar in the graphical user interface and on touchscreen devices.



To make scroll bars easier to use without a wheel-mouse or “edge scrolling”, the scroll bar can become wider when the mouse pointer is positioned very close to it. Diagram 2 shows the wider scrollbar.



In diagram 3, the dashed lines show roughly the area where the mouse pointer could be positioned in order to trigger the widening of the scroll bar. When the mouse pointer is moved away from it, the scroll bar becomes thin again.



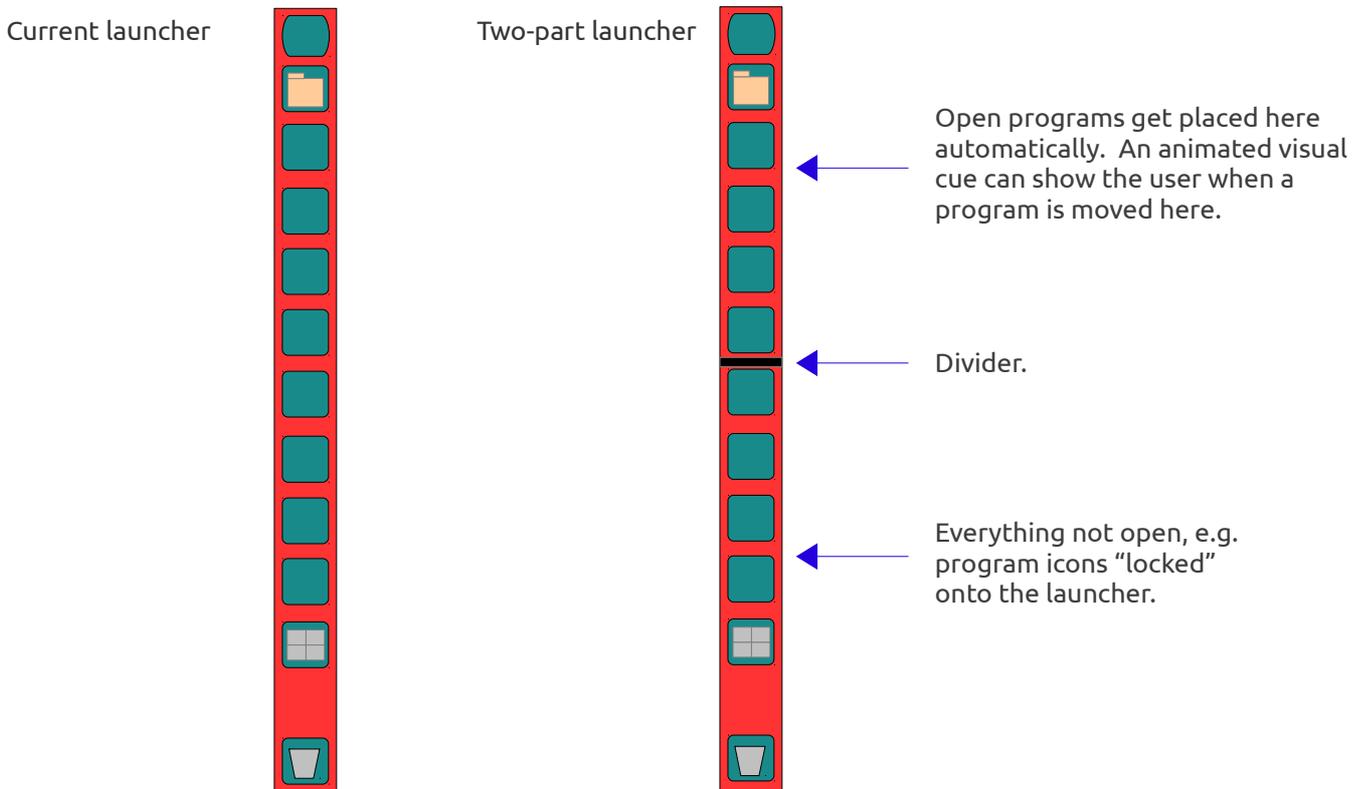
Mouse pointer on all scroll bars

Whilst the left mouse button is held down on a scroll bar, the mouse pointer can be “locked” onto the scroll bar, to prevent the mouse pointer from “falling off”. This includes both vertical and horizontal scroll bars.

Unity launcher

If you have several (or many) different programs open, you must traverse the *entire* launcher in order to find the open program(s) you wish to click on. A “two part” launcher would solve this problem and having each part automatically resize when required would increase the vertical height available in order to facilitate navigating. When no programs are open, the launcher reverts back to its original look – without the divider.

See how this could work and improve the current launcher.

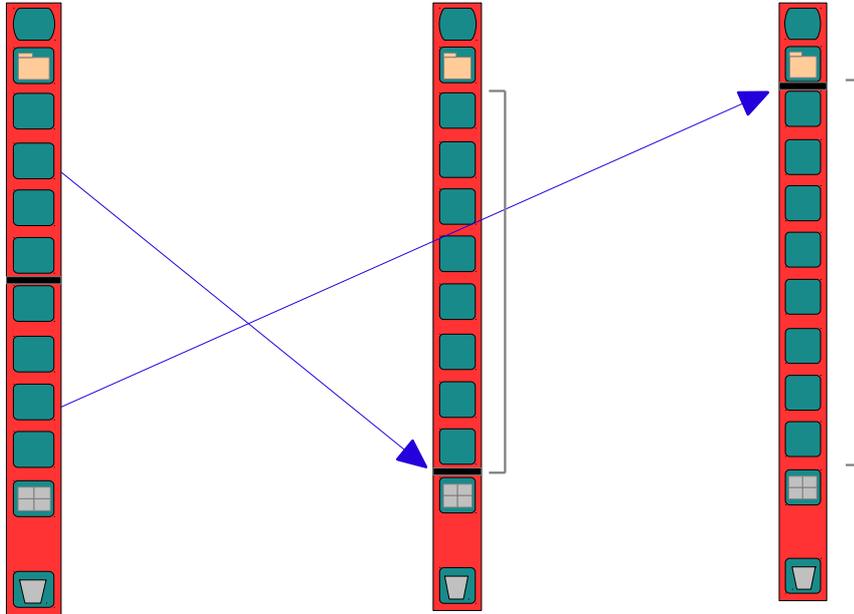


When the mouse is moved into the top or bottom half, the vertical height of the launcher is changed to reveal more icons. Diagrams 2 and 3 shows what this looks like when the mouse pointer is moved into the top half or bottom half of the launcher respectively; the divider shows the height change.

1) Top half has open programs. Bottom part has everything else.

2) The mouse pointer in the top half expands the launcher, in order to show its icons.

3) Likewise, the mouse pointer in the bottom half expands the launcher to show the icons there.



If the number of icons exceeds the expanded vertical height in either the top or bottom half, the launcher can scroll up or down like it currently does, following the mouse pointer.

Four icons remain permanently in their same places: Dash Home, Home Folder, Workspace Switcher and Rubbish Bin.

Show desktop

Unity needs a single button which, when clicked once, minimises all open windows to show the desktop. Not all users know the keyboard shortcuts and it's easy to forget keyboard shortcuts. This button needs to be permanently available in Unity and within easy reach.

Up and down indicators could be added to the launcher. These would show the user that the launcher has more icons than already showing, and the rest can be revealed by moving the mouse up or down onto – or very near to – the semitransparent “arrow” indicators. These indicators, shown on the right, could **always** be visible **whenever** there are more icons than showing, even when the top or bottom half of the launcher has expanded taller.



Input focus

The program with input focus should have its icon on the launcher appear “pressed inwards” to make it immediately visible. If a program generates a dialogue which appears above all other windows on the stack, this program should be the one on the launcher which looks “pressed inwards”, to show the user which program has generated the dialogue. Only one program at a time can ever have input focus.

When the password dialogue appears, always give input focus to the text box of the password dialogue, instead of requiring the user to first click into the text box.

Programs requiring immediate attention from the user

A program may require the user's attention. A perfect example of this is the Update Manager. When the Update Manager appears, the user may not notice it and the user could shut down Ubuntu without ever installing the updates! If a program on the launcher requires the user's attention, gently “blink” or “wiggle” the icon until the user clicks on it. Note: if the user has chosen to hide the launcher, *slightly* move the animated icon to the right, enough for the user to see it and keep it showing until the user unhides the launcher.

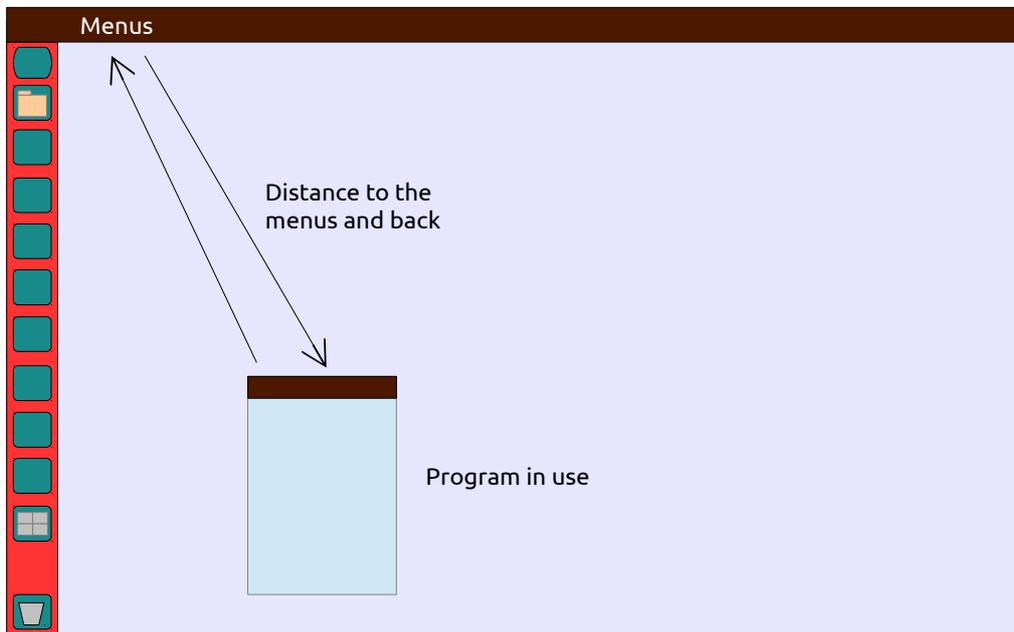
None of this applies whilst fullscreen is in use e.g. a game being played in fullscreen.

Alternative to focus stealing

As mentioned above, if the program requires the user's attention, show this on the program's icon in the launcher to draw the user's attention to it. This method could be used instead of focus stealing, except for critical system messages which the user must act upon immediately.

Distance to the Unity menu bar

Almost all programs' menus are placed in the top bar of the screen. This is good for consistency, but when programs are situated lower down on the screen, it causes confusion. Also, this setup requires moving the mouse long distances between the program and the menu bar – then back again!



Possible solutions: if the window is maximised or touching the top bar, then place the menus in the top bar. Otherwise, to save a line of screen space, only show the menus in the program when the mouse pointer is close enough i.e. directly below the title bar of the program's window. Alternatively, put the menus back into the programs and only use the menu bar at the top of the screen when a program is maximised.

Menus and the HUD

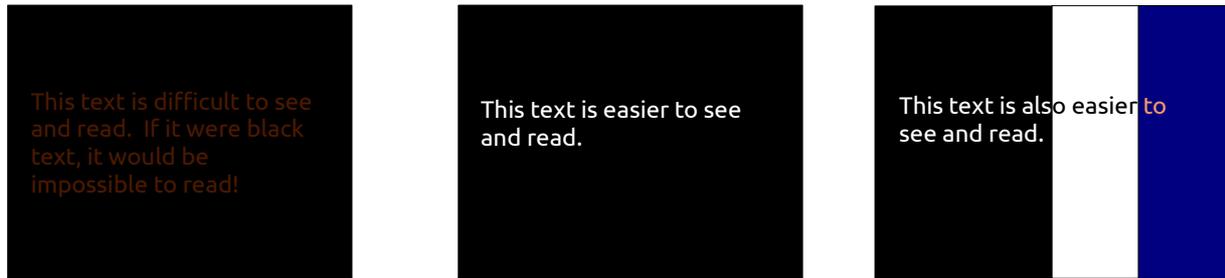
The Head Up Display (HUD) has not yet replaced traditional application menus, but the HUD looks very promising.

Consider allowing the user to click on more than one menu entry in the HUD, without the menu closing every time an entry is clicked on. This could be implemented in ALL menus (not just the HUD) by allowing the user to click on the menu entries with the right mouse button, instead of the left button, each time. If right-clicking a menu entry updates the menu, the menu can be redrawn to reflect this. But left-clicking a menu entry should still close the menu.

If the program with input focus also uses the Alt key for something else, the HUD could accidentally be activated at the same time. To avoid this, consider only activating the HUD with the Alt key when the mouse pointer is in the top bar of the screen. A tooltip could appear when the mouse pointer is in the top bar, reminding the user that pressing the Alt key will show the HUD menus.

Background colour affects foreground text

Applying “reverse video” would always make the text visible, regardless of the background colour. Other graphical elements could benefit from reverse video. The text in the 3 diagrams below illustrate this idea.



The diagram on the right shows how reverse video makes text easier to read over more than one background colour. It's similar to the text on desktop icons being against a background “wallpaper” with many different colours. The text rendering could take into account the background colour in a similar way to the method used in the “RISC OS” operating system, a method called “background blending”. [Click here to view the webpage](#) showing the text rendering (link may not work in Evince PDF viewer; you could right-click on it, copy the link address and paste in a web browser).

Workspace Switcher

Switching to different workspaces and back in Unity is more cumbersome compared to switching workspaces in the GNOME desktop environment, used previously in Ubuntu.

In Unity, you cannot see which workspace you're already in and you can't see – at a glance – which workspace(s) have open programs. Unity should adopt the workspace design used in the GNOME desktop, which does not suffer from these drawbacks. The Workspace Switcher in Unity should be within easy reach so the user can instantly click on a workspace to access it. By adopting the GNOME workspace switcher and putting it on the top bar, in the middle area, this approach would have several benefits. It would be easier to access workspaces, only require a single mouse click to switch workspaces, it would allow the user to see which workspace(s) have open programs and show which workspace is in use.

If implemented, this would also benefit small netbook screens, because you are now offering the user extra 'screens' in easy reach. Maybe add 2 workspaces by default instead of 4, due to space limitations.

Unity launcher and screen resolutions

The size of the Unity launcher icons could be automatically changed for different screen resolutions. Bigger icons used for higher resolutions and smaller icons used for lower resolutions.

Default applications

Make it easier for the user to find the settings for the “Default Applications”. Why is it hidden in “System Settings” > “Details” (for system information) rather than being allocated as a single entry labelled “Default Applications” in the system settings?

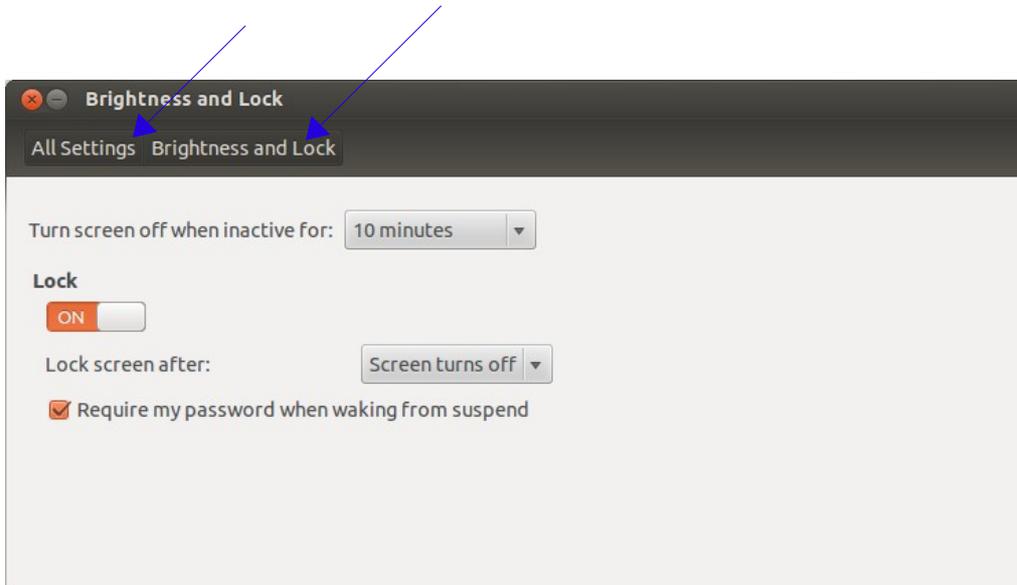
Mouse cursor/pointer auto-speed

Auto-set the mouse/touchpad speed to work comfortably according to the screen resolution. When the screen resolution is higher e.g .1920x1080, automatically set the mouse speed slightly faster. If the screen resolution is set lower, adjust the mouse speed accordingly. This would save the user having to make fiddly adjustments to the “Acceleration” and “Sensitivity” settings for the mouse (or touchpad), in order to make the speed more comfortable ergonomically.

Navigating the System Settings

A very simple change is needed: distinguish between “All Settings” and the category that's open. The screenshot below shows how the 2 “buttons” could be improved here.

These 2 “buttons” look too similar. Consider making the category button a different colour to “All Settings”. In this example, the category is “Brightness and Lock”.



An alternative suggestion is to use a “back” button, like that used in a web browser, to return back to all the settings.

Read about more “user-friendly” operating system design features [on this webpage](#).

Work created by David, user “turbolad” on the Ubuntu Brainstorm website.
E-mail: turbolad995@yahoo.co.uk