Spring 5-10-2009

Alternative Task Bar: A Usability Study

Jordan Cote
University of Connecticut - Storrs, jordocote@gmail.com

Follow this and additional works at: http://digitalcommons.uconn.edu/srhonors_theses

Part of the Graphics and Human Computer Interfaces Commons, and the Other Computer Sciences Commons

Recommended Citation
Alternative Task Bar - A Usability Study

Jordan Cote
Professor Russell
Senior Design Project - Honors Thesis
24 April 2009
University of Connecticut
I. Introduction .................................................................. p. 1

II. History and Current State of the Task Bar ............... p. 1

III. Issues With Current Task Bars ................................. p. 4

IV. Proposed Solution ....................................................... p. 6

V. Implementation ............................................................ p. 7

VI. Survey Overview ......................................................... p. 10

VII. Survey Results ......................................................... p. 11

VIII. Conclusion .............................................................. p. 13

IX. Works Cited ............................................................ p. 14
I. Introduction

The task bar has been an integral part of Graphical User Interfaces for over a decade. They provide computer users a way to launch and manage applications, which is a necessary and extremely common action. An inefficient or difficult-to-use task bar can, over time, accumulate into a substantial amount of frustration and lost productivity. This paper focuses on problems with current task bars, and a proposed solution for those problems. The implementation of that solution will be briefly discussed. Following that, an analysis on the effectiveness of the proposed solution will be made, based on a usability survey.

II. History and Current State of the Task Bar

The first task bar was introduced in Windows 95, in the year 1995. It resided along the bottom of the screen, and featured the “Start” button. This single button provided the means to launch applications via a series of embedded menus. Next to it, buttons representing open windows were provided. Finally, the “system tray” represented active applications which do not have associated windows. Through the evolution of the Windows task bar, it has gained theming capability to make it more visually appealing. The visual differences between the Windows 95 version and the latest version are readily apparent in figures 1 and 2. The first image is from Windows 95, and the second is from Windows Vista, released nearly 12 years later.

Figure 1. Windows 95 task bar (1995)

Figure 2. Windows Vista task bar (2007)
We can see from Figure 2 that there is a new feature on the task bar in Windows Vista (originally introduced in Windows 98), which is named “quick launch.” This feature provides direct links on the task bar for frequently used programs. Since then, however, the functionality of the task bar has not changed significantly. The organization of the Start menu has changed slightly, but not the task bar itself.

There have also been other task bar solutions. Apple’s Mac OS X Dock application is a type of task bar which is designed to be aesthetically pleasing. Dock focuses exclusively on launching certain applications, and monitoring open windows. It was originally released in 2001. The 2007 version (shown below, figure 3) does not differ much, apart from the “Stacks” feature (see figure 4). This feature allows for small menus, previously not featured in Dock.

Figure 3. Mac OS X Dock (2007)

Figure 4. Mac OS X Stacks (2007)
Even more incarnations of the task bar may be found on the various distributions of the Linux operating system. Most commonly, Linux variants use desktop environments Gnome or KDE, each of which have different task bars. Gnome features the Panel application by default. It is not themed at all, and appears as a grey box. The functionality is very similar to that of the Windows 98 task bar. The major difference is that it is split into two separate bars. The top bar (figure 5.a) is for application launching via menus and “quick launch” style icons. The bottom bar (figure 5.b) is for managing open windows and desktop control.

![Figure 5.a. Linux Gnome Panel (2005) - Top](image1)

![Figure 5.b. Linux Gnome Panel (2005) - Bottom](image2)

The KDE variants use the Kicker application by default. It closely resembles a newer Windows task bar, sporting a 1:1 correlation of features such as a menu button for launching applications, “quick launch” style buttons, active window monitoring and a system tray area (see figure 6).

![Figure 6. Linux KDE Kicker (2002)](image3)
**III. Issues With Current Task Bars**

It is readily apparent that the majority of task bars bundled with popular operating systems have much in common. Apart from the Mac OS X Dock, task bars have changed very little over the past 14 years. In stark contrast, many other aspects of computing have changed drastically. Take, for example, Microsoft’s web site ([http://www.microsoft.com](http://www.microsoft.com)). In 1995, it looked as depicted in figure 7:

![Figure 7.a. Microsoft web site (1995)](image)

When compared to the current Microsoft web site in figure 7.b, we can see that many advancements have been made.
The web site has changed considerably not only in looks, but also in functionality. And yet, the task bar has not added any major functionality since 1998. Could it be the case that Microsoft had designed the task bar so well in 1995, that it barely needed to be changed for over a decade? It is also possible that because users have become accustomed to it, the demand has not been enough to give rise to change. This study attempts to answer this question by proposing an alternative. Rather than relying mainly on an embedded menu approach, a style more similar to that of Dock is proposed.
IV. Proposed Solution

The Snoek task bar is developed in an effort to answer the questions above. Its goals include utilizing the power of modern computers in order to help the user experience. Efforts were made to make the program simple enough to use so that it requires virtually no learning curve for someone who uses computers. The program was developed for Linux with Gnome desktop environments. The reason for this is because it seems to have the most uninteresting task bar, and yet it is used by the most popular Linux distribution for home users, Ubuntu.

Snoek was developed based on the concept of a task bar that changes shape depending on the user’s actions. A common complaint about task bars is that they take up space on the screen, although they are only used intermittently. By taking up space only as needed, or when the user expresses interest in the task bar, this issue is abated. Another problem with Gnome Panel in particular is that it is visually unappealing. Contrasting this, Snoek uses an OpenGL graphics system to draw itself. This results in the developer having no restrictions on what can be done visually with the task bar. We leveraged this ability in order to draw smooth curves, create lighting effects and perform animation.

OpenGL has typically been used for games, image rendering programs and computer aided design. However, its use with standard desktop programs is becoming more prevalent. OpenGL provides a much more powerful interface than that of standard frameworks which provide classic window elements such as buttons, labels and icons. Also, as higher-end graphics cards become more ubiquitous, there is no reason they cannot be used to deliver a more user-friendly experience. We were also able to extend our application to be of an arbitrary shape, rather than the standard box shape to which OpenGL applications and widgets are normally confined. Therefore, our application is able to integrate seamlessly with the desktop environment while providing high-end graphics.
V. Implementation

The Snoek task bar uses this graphics framework to provide features that a normal task bar cannot. It is through this graphical and functional enhancement that the task bar will stand out from others. One such enhancement is shown when the user moves the mouse cursor over the task bar. A light source illuminates the icons and background. This light source follows the mouse cursor, and helps to focus the user’s attention.

Another visual feature is the shape of the bar. It is initially of similar width to a normal task bar. However, it resides in the corner of the screen with one arm stretching vertically and the other arm stretching horizontally. The arms are also rounded off at the ends. The arms can be of arbitrary length, as determined by the user. Resizing is achieved by clicking and dragging the end of either arm. When this happens, more or less icons are displayed in order to fully use the space that the task bar takes up on the screen. This presents an improvement over task bars which take up the entire bottom of the screen, whether they use the space or not. The space utilization and resizing features can be seen in figures 8.a and 8.b.
Figure 8.a. Snoek task bar, normal mode

Figure 8.b. Snoek task bar, resized
Apart from resizing, there is another way that the Snoek task bar changes shape. When the user moves the mouse cursor into the corner of the screen, the task bar bulges outward. This means that the endpoints for the arms are set as anchors on the edge of the screen, while the actual task bar grows to become a half-semicircle shape. With this larger surface area, more icons may be displayed. This functionality was implemented to provide access to applications which are not available in the standard, collapsed, mode.

The shapes are dictated by six 4x4 arrays of points for each mode. Initially, only selected control points are specified, then interpolating functions fill in the non-control points. Control points are those necessary for the desired shape. For example, four points along a curve must be specified, as well as corners of shapes. The six shapes are the top end, the vertical strip, two corner halves, the horizontal strip, and the right-hand end. The shapes are then joined at their junctions to eliminate tearing between shape segments.
VI. Survey Overview

In order to test this idea, a survey was conducted. Participants were asked to use the Snoek task bar, without being given any instructions. By not telling the participants how to use the program, we could observe how long it took for users to learn the controls. Because the program is meant to be intuitive, this information is valuable because it would help answer our previous questions about task bar usability. It can also pinpoint drawbacks of this concept.

While participants were not told how to use the program, questions were posed to help assess the program’s usability in certain situations. For example, participants were asked how they would go about resizing the task bar. The purpose of the question was to discover if the controls of the application are intuitive.

There were eight survey questions given to the participants, after they had finished using the program. The questions were aimed to answer questions about the concept’s usability. Questions were asked to also find out how advantageous the OpenGL graphics architecture was to the typical end user. These questions are presented in figure 10:

<table>
<thead>
<tr>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>How easy is the program to use, on a scale of 1 to 10? (with 10 being very easy)</td>
</tr>
<tr>
<td>How long does it take to learn the key features? Was it confusing?</td>
</tr>
<tr>
<td>Do the enhanced graphics make you more apt to use the program?</td>
</tr>
<tr>
<td>Do the graphics / animation help with program functionality / ease of use?</td>
</tr>
<tr>
<td>Does the program behave as expected?</td>
</tr>
<tr>
<td>What features were expected / desired, but not present?</td>
</tr>
<tr>
<td>What did you like most / least about the program?</td>
</tr>
<tr>
<td>Which of the following would you prefer for your own computer:</td>
</tr>
<tr>
<td>Windows Start Bar, Mac OS X Dock, Linux Panel, or this one?</td>
</tr>
</tbody>
</table>

Figure 10. Survey questions
VII. Survey Results

The survey was given to six participants. The salient features of the result set are that Snoek was well received, the OpenGL graphics make the program more appealing, and that most people prefer whichever task bar they use most. Participants expressed that they found the application easy to use, and found the controls intuitive.

The first question simply asks how easy the program was to use, on a scale of 1 to 10, with 10 being easiest. For this question, responses ranged from 8 to 10 with the mode being 8. The second follow up question indicated that all of the participants learned how to use Snoek quickly. One response noted that it was surprising at first when the shape of the task bar changed, but after the first time it was not confusing.

The third and fourth questions ask about the program’s appearance and use of OpenGL. Users seemed to enjoy the graphics. One person said that Snoek was “fun / easy to use.” Another expressed that they like that the program icons are illuminated by the light source which follows the mouse cursor. Some participants interpreted the term ‘graphics’ to mean the program icons themselves, and so some of the responses were not particularly useful. With this interpretation, one mentioned that having the icons available without having to click on a “Start” button is easy. Other users agreed that the changing modes help from a functional standpoint. In other words, the two modes helped to visually organize the programs.

The fifth question inquires about whether the actions of the program were expected or not. The answers indicate that the program did behave in an expected way. Supporting this fact, participants were always able to resize the application when asked. They were able to do this typically within 5 - 10 seconds.
The sixth question asks about what features could be added to Snoek. The overwhelming top request was for text labels for the application icons. Because of this demand, text labels are being implemented in the application.

For the seventh question, participants opined their favorite and least favorite aspects of the application. Responses for favorites included the resizing feature, the visuals, and the spatially compact form. By observing the participants using the program, it was found that people most enjoy the resizing feature. Responses for least favorite included the lack of description labels for applications, icons being too small, and the resizing grab area being too small. For further development, it would be a top priority to have icons change size based on user interest. Also, the grab area has since been enlarged, in order for users to more easily resize the task bar.

The last question simply asks which task bar the participant would prefer to use. The majority preferred Mac OS X Dock (3 people), followed by Windows Start Bar (2 people) and Snoek (1 person). One noted that if they were to switch to a PC they would use Snoek.

The result of the survey shows a positive reception of the concept for end users. The OpenGL graphics are well appreciated, and help with functionality as well. Controls for the application were easy to pick up, and users found the program intuitive. There were a few aspects that were found to be undesirable, but these issues were largely due to the program being in an early developmental state at the time of survey.
VIII. Conclusion

An alternative to the standard task bar has been proposed, and a usability survey has returned positive results. The idea of a shape-changing task bar seems to be a viable option for today’s operating systems. Because Mac OS X has a generally well-liked task bar, it is not the ideal target platform. Windows’ proprietary nature also discourages our efforts on that platform. However, the Ubuntu distribution of Linux is proud of being the most user friendly distribution but has the most uninteresting task bar. Also, because of its open source nature, it would be the best platform to deploy this task bar.

The integrated OpenGL approach was well liked by survey participants. The fundamental ideas were verified to work as intended. With further development, our study shows that Snoek could compete with the other major task bars.
IX. Works Cited

