

Augmenting Mac OS X window behavior and appearance

An Example Assignment

Design Problem

"I need you to create some believable (and do-able) design ideas for augmenting Mac OS X windows with new functionality that will set them apart from all other GUI windowing systems. By the way, you'll have that in my inbox by the close of business today. "

My Design Approach

After reading the design problem, I begin by considering the fact that the word "augment" was used rather than "enhance". Sounds like we need to provide some additional functionality, not just tweak things. We want these never-seen-before features to be innovative, but they need to be solid designs that don't undermine the existing windowing paradigm.

Since our team has control over the appearance, behavior and prototyping, let's take advantage of the somewhat clean slate that Mac OS X provides, since it's based on NeXT technologies. Also sounds like we have license to be innovative to support this major release. We should be able to tap some of new technologies (for Mac owners) inherent to Mac OS X for this problem. The primary ones that come to mind are transparency via the alpha channel in Display PostScript, incredible horsepower of the G3 to do fast animations and graphics processing, and the yet untapped 3D capabilities.

As usual, I'll write-up these design enhancements and include some hand drawn sketches or mocked-up Photoshop screenshots. Although, for this one, I'd really like to provide a few working prototypes in Director and/or hack some code to get my basic ideas across.

Getting started, I sketch out on paper the rough concepts. As I work through these one-by-one, I try to visualize them first. The vision is there, but I need something tactile to confirm. The feel and timing of the interactions is important, so rather than go straight to paper prototypes or Director, I'll hack something together quickly. Since I already have prototyping framework, I code up one of the simpler interactions to see how it feels.

After getting some initial idea of that it might really fly, I begin to work out what's needed for a Director prototype that could be used to get some feedback from others. I'd use the code prototype, but it turned out a bit too rough interaction-wise, and since that's the focus here, I opt for the Director mockup to get the full effect across. Should have started with Director in hindsight.

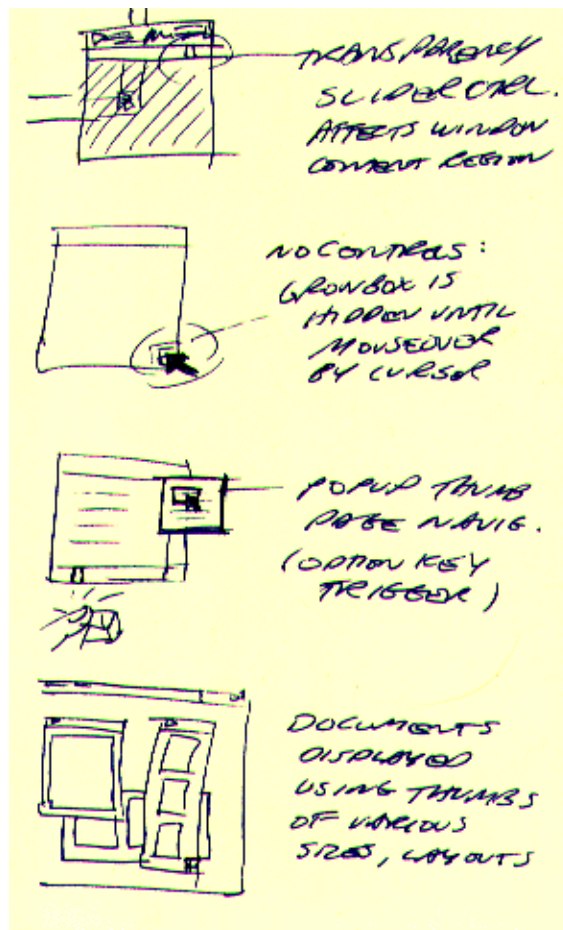
Before getting any real feedback, I check with Engineering to see if the underlying technologies will be ready in time and if what I'm doing is reasonable coding-wise. Don't want to sink the schedule for these features alone. The next step is to run through these concepts with others on my team and unsuspecting victims. I don't lead them, just gauge their reaction to it on first attempt and then after they've discovered the secret. The important aspects of the software interaction begin to surface, and a plan for refining the software interaction takes shape. Next step is to collaborate with my teammates to get additional insights and verify the design.

My Design Concepts

Here are the major points of my designs:

Augment the existing windowing paradigm, don't re-create it (for now...)

Listing the essence of the designs help to crystallize what's really important, and remind me to keep things as simple as possible. As I look over the sketches I've made, it seems that a few are processor intensive and the others require the ability to create thumbnails of pages, which of course means these need to be generated quickly and cached. None have compromised what has been established as the Mac Human Interface guidelines except #2. And even that one is somewhat open to interpretation, since the functionality is there, just hidden until a rollover.



1. Transparency slider for windows

All window types will have the capability to go from completely opaque (normal) to transparent. This allows people to see the underlying documents more clearly than tool palettes, etc.

2. No controls visible by default

Along the same lines as transparency, it makes the focus of the work the data, not the user interface. The standard window controls are displayed temporarily when the mouse rolls over the standard areas.

3. Page Thumbnail Navigation

Similar to the Acrobat thumbnail navigation controls, this popup window is associated with the underlying window. Allows quick movement or scaling of the doc. Triggered by a modifier key or physical mouse control.

4. Finder Thumbnail Views

Add the ability to View by Thumbnail to the Finder. Store thumbnail previews of each page in the doc, so it can be browsed without opening it in the native application.

Figure 1. notes on the design concepts

Design Concept Problems & Solutions

Here are the major problems that are being posed and how they're being addressed:

1. Transparency slider for windows

All window types will have the capability to go from completely opaque (normal) to transparent. This allows people to see the underlying documents more clearly than tool palettes, etc.

Based on Xerox MagicLens idea of using palettes more effectively. Need to check on patent issues with this one.

Potential Problems

- Big change in window behavior, so some experimentation needed by users
- Z-ordering issues have to be dealt with (i.e. can you click through a trans window?)
- Need to associate many windows to a single transparency slider (i.e. the app's slider)

2. No controls visible by default

Along the same lines as transparency, it makes the focus of the work the data, not the user interface. The standard window controls are displayed temporarily when the mouse rolls over the standard areas.

The main idea is to clear the user interface away from the underlying data, which is really the important part. No known patent issues, as this is used on the Web and multimedia projects.

Potential Problems

- Requires people to rollover existing areas to uncover the standard functionality
- Windows look odd at first sight, need to explore whether the titlebar needs to be visible
- Without some type of HUD super-imposed on the content area, scroll position isn't shown

3. Page Thumbnail Navigation

Similar to the Acrobat thumbnail navigation controls, this popup window is associated with the underlying window. Allows quick movement or scaling of the doc. Triggered by a modifier key or physical mouse control.

Could go along with #2, since there are no visible controls for page navigation, this could be used as a heads-up display navigation control. Could be drawn using slight transparency. The small size of the thumbnails would make page navigation and zooming preview quick.

Potential Problems

- Requires thumbnails to be generated for each page in a document.
- New interface control for those not familiar with Acrobat-style thumbnails

4. Finder Thumbnail Views

Add the ability to View by Thumbnail to the Finder. Store thumbnail previews of each page in the doc, so it can be browsed without opening it in the native application

Could go along with #3, since thumbnail generation is required for both. This seems to be a natural extension of the View By... mechanism in the existing Finder. The twist is being able to view the pages of a document, not just the icons of the documents in that folder. Same rules apply after you get into.

Potential Problems

- Requires thumbnails to be generated for each page in a document.
- Performance of the Finder could be adversely affected.

Time and Tools Used

It took approximately six hours (including interruptions) to complete this exercise, which included considering the problem, sketching out some designs on paper, hacking a prototype in C++, starting a Director prototype, rethinking the solution, write-up of this document in Microsoft Word.