by Dave Parfrey
10/1/92

Introduction

In my mind, this column should discuss topics that the majority of Nautilus subscribers want to hear about in the world of programming. However, since I don't have the first clue what any of those topics are, I have no choice but to begin by writing about what's foremost in my mind. (Well, actually, I've got lots of choices, but let's not quibble.) Since I'm basically a pretty focused person (as most programmers are), I've decided to write about my current project.

This is currently envisioned as a multiple-issue discussion, but if you've got different ideas about what I should be writing about, let me know and maybe I'll change directions midstream (or maybe not. After all, it's **my** column!). As long as I'm digressing (Already! Not a good sign.), I'll be dropping hints about future columns along the way, so your reactions to them can push those ideas up (or down) in priority. Finally, since Nautilus is, after all, a multimedia magazine, Ty and I hope to be experimenting with ways of differentiating these columns from the type you'd read in *Doctor Dobb's Journal*, *Windows Tech Journal*, or *PC Magazine*. Since animated **if**-statements don't seem too exciting to me, your ideas are welcome.

My current project

I'd like to give you a detailed description of my project, but I can't. My client won't let me. Competitive advantage and all that rot. When it's finished, I'll try to include a demo of it in a Nautilus issue. But for now, a general description will have to do. It's not really all that critical about the topic I'm aiming at, anyway.

This application is a corporate training application. It's **not** a CBT (computer-based training) application, at least not in the standard definition of that overworked term. (I define a CBT application as an application that entertains students without teaching them anything. A **bad** CBT application doesn't even entertain them!) This application trains by repeatedly asking the student questions in timed 1-10 minute sessions, sort of like flash cards. It's based on sound educational theory (I'm told) and significantly increases retention. Version 1 of this application is currently being used, for example, to teach insurance sales people details about the different policies their company offers.

I wrote the first version of this application in Clipper. Hey, don't leave already! I did it as a prototype! It had to be done quickly! I have a million other excuses! I'm writing version 2 in Borland C++ using the Zinc Interface Library. If that didn't bring you back, nothing will.

Actually, Clipper is a pretty good environment to get a lot of work done quickly, and the Clipper version works very well. But the client keeps coming up with new wrinkles like different types of questions to ask and more detailed statistics. They want mouse support, graphics, and even (gasp!) Windows support. The Clipper version's been bulging at the seams for awhile, so I convinced them it was time to recode it in C++.

Ideally, I needed to find a screen library that would support DOS text mode, DOS graphics mode, and would also run as a Windows application, **all from the same source code**. Why use the same source? Well, the client would eventually like to branch out to the Macintosh and maybe even UNIX and VMS, making portable code very nice. Other contributing factors were the client's (unfortunately) limited budget and the fact that I really hate coding the same application over and over and ...

The Search

A lot of people on the CompuServe forums have been facing this same problem. In addition to scanning catalogs and magazines, I started asking questions and "lurking" on several forums to learn about the different products. The forums I used were BCPPDOS and BCPPWIN, mainly because, as a confirmed Borland C++ fan, I already hung out there. In a short time, I came up with the following products to consider:

Zinc Interface Library, by Zinc Software Incorporated, 405 South 100 East, 2nd Floor, Pleasant Grove, Utah 84062. Phone: (801) 785-8900 FAX: (801) 785-8997 BBS: (801) 785-8996.

MEWEL, by Magma Systems, 15 Bodwell Terrace, Mullburn, New Jersey 07041. Phone: (201) 912-0192 FAX (for orders only): (201) 912-0103 BBS: (201) 912-0668. MEWEL also has a forum on CompuServe (PCVEND).

WNDX, by The WNDX Corporation, Suite 305, 1550-8 St. S.W., Calgary, Alberta T2R 1K1. Phone: (403) 244-0995. **

zApp, by Inmark Development Corporation, 2065 Landings Drive, Mountain View, CA 94043.
Phone: (415) 691-9000 FAX: (415) 691-9099 BBS: (415) 691-9990

Demos of these products are included in this Nautilus disc.
[****ED. NOTE**: WNDX would not allow their demo to be carried on this disc. However, you will find demos for the other listed products in the NEMO directory.]

I also considered (and rejected) Borland's ObjectWindows (OWL) and Turbo Vision. OWL is Borland's Windows-based applications framework and Turbo Vision is their DOS-based framework. I own and have used both. I would have used OWL in a heartbeat, but that wouldn't have helped me in DOS at all. Turbo Vision is OK, but it operates in DOS text mode only, and I feel that the primary operating environment for this application is DOS graphics. DOS text mode will be used only by those users with "antique" PC's without a graphics card. (I can freely insult those users here, since Nautilus users **obviously** don't fall into this category.)

The following off-the-cuff comments on these products don't really do them justice. I'm simply stating why I didn't find them appropriate for my application. Please check out the demos, because they all have definite advantages. I may go into more detail about these products in a later issue, if enough people are interested.

I really tried to like MEWEL. On the surface, it had it all. I could write my application as if it were a Windows app. I could even use OWL if I wanted, because MEWEL can compile the OWL code into a DOS app! It ran in both DOS text and graphics modes. But the MEWEL DOS applications were **ugly** (at least in their demos. The MEWEL folks tell me you can customize the DOS look at the expense of Windows compatibility.) DOS graphics mode was slow on my machine (a 486/33!) and looked more like the DOS text version than the Windows version. In my opinion, MEWEL will fit best for a Windows app that has a few users that just **have** to have a DOS version.

zApp uses MEWEL for its DOS version, so it really didn't fit my needs. zApp imposes its own application framework and messaging scheme (as does Zinc), and it seems to have some fans on CompuServe. But I've already gone through the Windows and OWL learning curve and I really have no desire to learn yet another applications framework, as least for Windows programming. Possible topic for another column: I'm tired of learning a new applications framework for each project!

I found out about WNDX fairly late in the process, after I'd already decided to use Zinc. It didn't seem to support DOS text mode, a major stumbling block for me. Also, on my VGA display, the default DOS graphics font was smaller than the default Windows font. In my opinion, it was too small.

Zinc did everything I wanted. It was attractive in all three modes. Zinc seemed like the obvious choice. Unfortunately, I knew too much about Zinc. I'd been thinking about the C++ version of this application from the minute I wrote the Clipper version (and that was close to 2 years ago!). In preparation, I bought version 1 of Zinc back then and played with it. It was promising, but full of bugs. I loved its class structure, particularly how it handled the text versus graphics mode problem with different Display classes (I'll describe Zinc's features next time). But it concerned me that both version 1 and even version 2 couldn't get scroll bars to work properly. They just acted weird sometimes, and this was in Zinc's own demo application! Also, the CompuServe forums were not kind to Zinc. Tirades from disappointed programmers scared me. In spite of these warning bells going off in my head, I felt I could make Zinc work, especially with their brand-new version 3.

Next issue: How is Zinc working out? What do I like and dislike about it?

Tip of the month: Try out the String++ class in this issue, written by Carl Moreland. I downloaded it from CompuServe (BCPPDOS forum). If you're tired of using strcpy, strcmp, and all those variations, you'll love String++, and it's free!