The Developer Business Report July 1995

AppleDirections

Inside This Issue

Errata

We regrettably omitted a byline from last month's issue. The article "OpenDoc—One Architecture Fits All" was written by *Apple Directions* Technical Editor Gregg Williams.

Apple News

Power Macintosh 9500 Offers Next-Generation Computing

Computer Adds PowerPC 604 Processor, PCI Bus, Open Transport for Under $5,000

The Power Macintosh computers you've seen so far were just the beginning. When Apple Computer, Inc., began its transition to the PowerPC processor architecture with the Power Macintosh 6100, 7100, and 8100 computers, it did so with compatibility in mind. The introduction of the Power Macintosh 9500 marks the beginning of a new, second-generation Power Macintosh product line—in which maximum performance was foremost in the designers' minds.

The Power Macintosh 9500/120 offers the following impressive list of features, all at a price of $4,999 (U.S. price only):

- PowerPC 604 processor, running at 120 MHz
- six industry-standard PCI (Peripheral Component Interconnect) slots
- 1 GB (gigabyte) internal hard disk drive
- 16 MB of memory, expandable to 768 MB
- 24-bit graphics accelerator card

please turn to page 9

Strategy Mosaic

Strategy: 1999

By Gregg Williams, Apple Directions *staff*

Part 1: The Evolution of Apple's Key Technologies

It's no accident that this column is called "Strategy Mosaic." Apple's strategy has always been a broad one, encompassing many technologies—too many, I thought when I named this column, to be covered in one article.

That statement is still true—I can't *cover* all of Apple's technologies. But for the first time, I'm going to attempt to *name* them and show how they relate to each other, your customers, and your products.

At each year's Worldwide Developers Conference, Apple Computer, Inc., management attempts to give developers the clearest picture it can of Apple's future directions, so that you can make informed business decisions about how your future products will incorporate Apple technologies.

This year was no exception. Ike Nassi, vice president of Apple Systems Software Technology, gave a three-part presentation of Apple's technology strategy from three perspectives:

- standards—how Apple technologies will evolve to create a more powerful platform for developers and a consistent yet scalable user experience for customers
- customers—how Apple technologies appear from the customer's point of view
- products—which Apple technologies are important to which categories of products

please turn to page 3

AppleDirections

Volume 3, Number 7

Apple Directions, the monthly developer newsletter of Apple Computer, Inc., communicates Apple's strategic, business, and technical directions to decision makers at development companies to help maximize their development dollar. It is published by the Apple Developer Periodicals group within Apple's Developer Press.

Editor
Paul Dreyfus (AppleLink: DREYFUS.P)

Technical Editor
Gregg Williams (GREGGW)

Business & Marketing Editor
Kris Newby (NEWBY.K)

Associate Editor
Anne Szabla (SZABLA)

Production Editor
Lisa Ferdinandsen (LISAFERD)

Contributors
Peter Bickford, Dave Curbow, Elizabeth Dykstra-Erickson, Alex Dosher, David Gleason, Don Norman, Kerry Ortega, Geoff Schuller

Manager, Developer Press
Dennis Matthews

Manager, Apple Developer Periodicals
Mark Bloomquist

Production Manager
Diane Wilcox

Prepress/Film
Aptos Post

Printer
Wolfer Printing Co., Inc., Los Angeles, CA

Editor's Note

Blown Away at the WWDC

In case you had to miss this year's Worldwide Developers Conference (WWDC), I thought I'd tell you about two of the best moments of the conference.

The conference was filled to overflow with solid technical and marketing information about Apple's existing and forthcoming products. For example, more than one of the developers I talked to at the conference was struck by the early, complete disclosure about Copland, Apple's next-generation version of the Mac OS. I was particularly struck by a couple of demos, one that I witnessed and one that I heard about and later viewed on videotape. They each showed just how inventive and innovative Apple remains, and why you really don't want to miss WWDC. And they both bowled me over.

The first of these demos was of Cyberdog, a forthcoming set of OpenDoc components that let users build their own interfaces for getting on the Internet. It showed just how easy Cyberdog components will make it for people to use the Internet; it was also the most compelling OpenDoc proof of concept I've seen to date.

Cyberdog engineer Steve Fisher set out to create an OpenDoc document that would provide access to information about Walt Disney Corporation's hugely popular movie, *The Lion King.* Steve proceeded to build the document, working without a net—that is, instead of showing a canned simulation, he worked with real, live OpenDoc software that could have gone into MacsBug at any moment.

Starting with a partially created OpenDoc document, he first drew a box inside the document, using one OpenDoc component. Next, he embedded a Disney news group from the Internet inside the box, this time using another OpenDoc component that provides connections to different types of Internet sites—World Wide Web pages, FTP sites, news groups, and so on. Now, the newly created box provided a connection to the Disney news group, and all the contents of the news group could be viewed by clicking in the box and scrolling though it.

After that, Steve selected a button from among a set of ready-made, so-called OpenDoc stationary buttons and dragged it to another part of the document. To distinguish the button, he took an illustration from the Scrapbook showing a character from *The Lion King,* resized it (using another OpenDoc component), and placed the illustration on the button. Then, using the OpenDoc Internet connection component, he connected the button to the *Lion King* Web page. Clicking the button then opened the page.

Finally, seeing a *Lion King* illustration on the Web page that he preferred to the one he'd used on the button, he copied the illustration from the Web page, cropped and resized it (again using the OpenDoc image-processing component), and placed the new illustration on the button.

Of course the audience went wild! He'd created a unique, creative interface to the Internet using only OpenDoc components, whose features immediately became available to him when he clicked on the pertinent parts of the document. It looked so easy that I thought even I could have done it, and—even better—it looked like something I'd *like* to do some time!

I thought I'd seen *the* demo of the conference, but then a friend told me about the session on QuickTime VR, Apple's new technology for building electronic three-dimensional environments out of standard 35-mm photographic images. Fortunately, through the magic of videotape, I was able to see it for myself.

Ken Doyle, one of the QuickTime VR engineers, was about to show features of the next version of QuickTime VR when his computer screen shifted from a three-dimensional image to (oh no!) MacsBug. The audience politely applauded, showing that they understood that the Demo Gods weren't being kind to Ken that day.

Or so it seemed. With perfect aplomb, he clicked on the MacsBug screen and it

peeled away to reveal a three-dimensional QuickTime VR movie of the inside of a Macintosh computer—with an animated version of QuickTime VR inventor Eric Chen walking through the circuitry banging on things with a mallet! Even on the video, you can hear the audience cheering, and at the end of the demo, the camera pulls back to show them giving the team a standing ovation.

Amid all the excitement, I realized I'd just seen dramatic proof that the next version of QuickTime VR would be able to embed animation into 3D environments, a feature that many multimedia developers have been clamoring for. Another new feature will let you put standard QuickTime movies into QuickTime VR environments. (For lots more information about QuickTime VR, see the articles on page 11 and 19 of this month's issue.)

Those were my two favorite moments from this year's WWDC. I'm sure some of you were in those demo halls and (I hope) share my excitement. For those of you who weren't there, you should plan to come next year. For all Apple developers, the Worldwide Developers Conference is very much the right place to be in mid-May.

Paul Dreyfus
Editor

Strategy Mosaic

Strategy: 1999

continued from page 1

This month, I'll cover the first perspective—the evolution of the Mac OS platform through various key standard technologies. In next month's column, I'll cover the last two perspectives. That way, whether you were able to attend the WWDC this year or not, you will know where Apple technologies are headed and how best to focus on the technologies that are the most important to you and your customers.

Given the problematic nature of predicting R&D schedules, the exact timing of some technology components is not known. However, it's safe to say that this Technology Mosaic (which is what I'm calling the five figures on pages 6–7) includes pieces that are three to four years out—taking us almost to the next decade (yikes!).

Overview

The figures on pages 6–7 show the different pieces of the Technology Mosaic:

- hardware and drivers
- operating system and user interface
- graphics, text, and multimedia
- communications and collaboration
- tools and frameworks

Each horizontal row of items shows how a given technology "space" evolves over time. Three conventions are very important here: An arrow pointing to the right means that the technology on the left will be carried forward. No arrow pointing to the right means that a technology has come to the end of its life. An arrow pointing to the left means that a technology is compatible with whatever came before it.

Each figure has four columns of items. The columns, read left to right, indicate the passage of time from now to the late 1990s. Here is an explanation of what each column represents:

- System 7.5—represents technologies present in today's Macintosh computers running System 7.5
- OpenDoc and '95 Power Macs—represents technologies that are present after the introduction of the Power Macintosh 9500 (Apple's first PCI-based Macintosh; see the story on page 1) or OpenDoc (to be introduced this fall)
- Copland—represents technologies that will be present in Copland, the next major revision to the Mac OS
- Gershwin—represents technologies that will be present in Gershwin, the next major revision to the Mac OS after Copland, due in the late 1990s

The five figures and the rest of this column, which tries to explain and expand on them, contain a lot of details. To avoid getting lost in the details, you may want to read only the italicized items in each section, then come back and read each section in its entirety. But be sure to read the final section of this article, "A Necessarily Complex Strategy."

Hardware and Drivers

- *The PowerPC processor family is the future of the Macintosh.*

The PowerPC 601 processor, which has been used in Power Macintosh computers to date, is only the beginning. The newly announced Power Macintosh 9500 uses the PowerPC 604 processor, and future PowerBook and desktop Power Macintosh computers (as well as the PowerPC Hardware Reference Platform computers due in late 1996) will use the PowerPC 603, 604, and 620 processors.

More important, David Nagel, senior vice president of Apple Worldwide Research and Development, announced at the WWDC that Apple will soon be making a total transition to the PowerPC architecture. He said that Apple will be introducing no new 680x0-based Macintosh models (though it will continue to sell existing models) and that by the end of 1996 at the latest, Apple will stop selling 680x0-based Macintosh computers entirely.

At the beginning, Apple sold Power Macintosh computers primarily to the high end of the market. Now it is selling them widely to the entry-level, education, and consumer markets and will continue to do so as Apple sells more PowerPC processor–based computers and fewer 680x0-based ones.

This means that if you haven't already, you should start creating Power Macintosh "native" products, and you should design future products and enhancements to

Apple Directions Online—August

The July issue of *Apple Directions* will be available on AppleLink, the Internet, and eWorld by August 15, at the following locations:

AppleLink: path—Developer Support:Developer Services:Periodicals:Apple Directions.

Internet: select Developer Services and Products at the location www.apple.com.

eWorld: in the Apple area of the Computer Center.

take advantage of the extra power available in Power Macintosh computers. So now you have both the tools and the unit volumes you need to make this transition worthwhile.

• *PCI replaces NuBus starting now.*

Apple decided that the many advantages of the PCI expansion-card architecture far outweigh those of NuBus™. Because of this, all future new computer models will use the PCI bus. (For background on PCI, see "PCI Cards for the Second Generation of Power Macintosh," on page 13 of the August 1994 issue of *Apple Directions*.)

• *FireWire and GeoPort are in your future.*

FireWire is a new serial peripheral bus architecture that will provide a high-speed alternative to today's ADB and SCSI buses. FireWire is needed for multimedia and other solutions that must transfer data more quickly than SCSI peripherals can. FireWire interfaces will be available soon and will be standard in the Copland time frame.

GeoPort is not only in your future, it's here now, in the form of GeoPort 2.0. (For more on GeoPort, see "GeoPort Telecom Adapter Kit Offers New Opportunities," on page 9 of the February 1995 issue of *Apple Directions*.) GeoPort is an extension of the Macintosh serial interface. It is—and will continue to be—the standard hardware interface to the outside world for telephony-based applications.

• *The Copland driver model breaks today's model but offers important advantages; however, today's PCI cards will, if designed properly, work with Copland.*

PCI cards, obviously, require a different kind of driver software than NuBus cards. If you write your PCI drivers according to Apple guidelines, those drivers will continue to work under the substantially different driver architecture of Copland. (In Copland, drivers run in separate, protected memory spaces and cannot make Toolbox calls.) Additionally, all I/O will be faster under Copland because all I/O code will be written as native code.

(For more on PCI, see the Apple book *Designing PCI Cards and Drivers* and the document "New Device Driver Model." You can find both of these on the June 1995 Developer CD, using the pathname Dev.CD Jun 95: Technical Documentation:PCI Information.)

Operating System and User Interface

• *The Mac OS will evolve to more complex tasking architectures, culminating in pre-emptive, memory-protected applications and symmetrical multiprocessing in Gershwin.*

An operating system that can multitask (that is, work on more than one task simultaneously) and multiprocess (divide a task among multiple processors) is more versatile than one that can't. The Mac OS does not currently offer either full multitasking or multiprocessing, but it will by the Copland time frame. The transitions must be slow for reasons of backward compatibility.

The Mac OS has had cooperative multitasking for well over half a decade. It also has the Thread Manager, which allows an application to run cooperatively scheduled processes, called *threads*, within a given application. Copland expands this to full tasks that run pre-emptively (which is more powerful than cooperative scheduling) and in their own protected memory space. Copland itself is a stepping stone to Gershwin, where entire applications run pre-emptively in their own protected memory spaces. By using the Thread Manager now, you will ease your transition to Copland later.

Multiprocessor support will take another path. Apple has announced an agreement with DayStar Digital on a new API (application programming interface) that will allow developers to write software that will take advantage of multiple processors—for example, software for a Power Macintosh computer with one or more coprocessor PCI cards added). This technology should be available by early 1996 and be compatible with future multiprocessor directions.

The Mac OS will gain more multiprocessing capabilities as time passes. The fullest form of multiprocessing is symmetrical multiprocessing (SMP), and the Copland microkernel already incorporates core elements of SMP support. Full SMP will become part of the operating system itself in the Gershwin time frame.

• *Macintosh Drag and Drop is as fundamental as the Clipboard.*

One of the most important aspects of user interaction is consistency: Interactions that are possible in one place should be possible in other places deemed to be similar.

Users overwhelmingly prefer dragging and dropping data to cutting and pasting it—I know I do. But dragging and dropping is useful only when both the source and the destination support it, and this limits its overall effectiveness.

Apple created the Drag Manager to make dragging and dropping powerful and as easy to implement as possible. So that dragging and dropping can be as useful as possible, Apple strongly encourages you to follow this rule: If your software implements the Clipboard, it should also support Macintosh Drag and Drop.

• *Apple Guide and AppleScript will evolve, by the time of Gershwin, into intelligent agents.*

This is an example of the synergy of Apple technologies—two or more technologies that combine to create an entirely new technology. In this case, the Apple Guide technology, combined with the ability to link applications together through Apple events and AppleScript, will lead to active assistance in the Copland time frame and, later, to "intelligent" software agents in the Gershwin time frame.

I've been talking for years about the desirability of "factoring" your application to respond to Apple events. If you haven't done so already, you really should do it as soon as possible. For the Mac OS to be successful in its goal of actively helping users get their work done, applications and OpenDoc parts must be able to share their abilities with other software through their support of Apple events and AppleScript. Software that is "deaf, dumb, and blind" to Apple events will be seen as less useful than software that can participate in helping users get their work done.

(By the way, if you've put off implementing Apple events because they run so slowly, that problem will disappear in the future. Apple engineers reported at the WWDC that they measured a Copland application executing Apple events at a rate of over 1500 events per second on a low-end Power Macintosh 6100/60—pretty impressive!)

• *The QuickDraw GX printing interface will become standard in the Copland time frame.*

Though QuickDraw GX has a lot to offer, Apple admits that you need to rewrite large parts of your software to use all its features. It is a minor task, however, to implement the QuickDraw GX printing model, which gives the user a much better human interface for printing.

For this reason, Apple has been recommending—and now

does so even more strongly—that if your software does printing, it should implement the QuickDraw GX printing model. You need to make the transition now, because QuickDraw GX will become the standard printing human interface in the Copland time frame.

• *PowerTalk will improve over time; parts of it—the PowerTalk key chain and catalogs—will be integral to Copland.*

Apple is working on making PowerTalk more useful (through a fuller-featured mail program and other improvements) and faster. Copland will make two parts of PowerTalk standard: the key chain—a streamlined human interface for file-access security—and catalogs—a unified technology and human interface for accessing collections of data.

• *Copland will also bring significant enhancements to the user interface, including new visual appearances, workspaces, and better ways of navigating through data.*

Copland adds major new human-interface elements to the Mac OS interface. These include new visual appearances (different "looks" for windows, buttons, menus), workspaces (which allow different users to save and recall their preferred configuration of a Mac OS computer), and different versions of the Finder for users at different levels of expertise.

Graphics, Text, and Multimedia

• *Apple continues to improve QuickDraw GX; in the Copland time frame, it will become the standard way of doing text, graphics, and printing on Macintosh computers.*

The architecture of Copland will make it possible to deliver a version of QuickDraw GX that runs even faster and occupies less memory. (In addition, the Copland version of QuickDraw GX will be implemented as a shared library, and shared libraries are not even loaded into memory until they are needed.)

• *Apple provides strong support for localizing your software through WorldScript. In the Copland time frame, Apple will add more ways to localize your software, including direct support for Unicode.*

No matter where the primary market for your software is, the rest of the world represents another large market for you—which is why Apple has always worked to make to make Mac OS software as easy as possible to localize.

You should study the entire strategy, then focus on those parts that are relevant to your company.

WorldScript continues to be one of the main technologies you need to use to prepare your software for localization, even if you aren't planning on doing localization now. Copland will provide more extensive support for localization through its direct support for the Unicode text-encoding standard. Copland will also introduce the concept of text objects, which abstract text encoding from the remainder of an application, thus making the application easier to localize.

• *QuickDraw 3D provides a cross-platform standard for 3D graphics and will become more powerful as time passes.*

QuickDraw 3D, which is a Power Macintosh–only technology, will ship with all Power Macintosh computers later this year and is slated to be available for Windows-based computers in the first quarter of 1996. (For more information, see "3D for Free—Apple Announces QuickDraw 3D," on page 16 of the March 1995 issue of *Apple Directions*.) QuickDraw 3D version 1.5 will be even faster in the Copland time frame and will support an improved rendering engine.

Since QuickDraw 3D will be available to you royalty-free on both platforms, there's no reason for you not to investigate what Apple thinks is the best 3D technology available for Mac OS–based and Windows-based computers.

• *QuickTime 2.0 and its recently announced enhancements, as well as QuickTime VR and Sound Manager 3.0, provide powerful tools for creating media-rich products.*

QuickTime is and will continue to be a powerful tool for delivering multimedia easily to both Mac OS–based and Windows-based computers. In addition to providing improved movie playback and other enhancements, version 2.0 adds MIDI support. This allows you to add music (using a set of sampled musical instruments from synthesizer manufacturer Roland) with virtually no effect on your product's memory footprint or performance.

Apple has also increased the usefulness of QuickTime by announcing two additional enhancements. The QuickTime Music Toolkit includes technologies that mean new development opportunities for you. With this toolkit, you can create Enhanced CDs that can play their music content in a regular CD player but can also be played in a CD-capable computer for an enhanced entertainment experience.

In addition, Apple has announced that it will incorporate Opcode Systems' Open Music System (OMS) into QuickTime. OMS is a music-industry standard for the control of MIDI-based devices. Through OMS, QuickTime movies will easily be able to control and interact with MIDI equipment, thus making QuickTime more useful for music and multimedia applications.

• *Sound Manager 3.0—use it.*

There's not much to say about Sound Manager 3.0—it works, it provides excellent sound-mixing capabilities, and it will continue to be the API that you use to manipulate sound.

QuickTime VR is an exciting new technology that allows you to view and move around in three-dimensional spaces. (For a first look at QuickTime VR, see "QuickTime 2.0 in the Limelight," on page 14 of the February 1995 issue of *Apple Directions*.)

Apple has just announced a drastic reduction in both the pricing of QuickTime VR tools and licensing (see the news item on page 11), making QuickTime VR a more attractive technology for delivering software to both Mac OS–based and Windows-based computers.

Since the scenery files that QuickTime VR uses are small by today's standards (under a megabyte), it can provide multimedia experiences without requiring a CD-ROM or large transfers of information. This presents some exciting possibilities, perhaps even viewing 3D environments across the Internet.

Communication and Collaboration

• *Open Transport provides a powerful new architecture for networking that maintains compatibility with today's network-aware products.*

Open Transport (which we've scheduled to cover soon in *Apple Directions*) is being introduced with the Power Macintosh 9500 and later PCI-based Power Macintosh computers, but it will be moved back to support 680x0-based Macintosh computers in the future.

Open Transport makes networking easier for customers and system administrators. It allows you, the developer, to write one network-aware application that will work with more than one network architecture, giving you a streamlined development process and a larger market for your applications. By writing to a single API, your software will work with AppleTalk, TCP/IP, and any networking protocol that supports the UNIX® STREAMS environment.

One important feature of Open Transport is that it is compatible with pre–Open Transport applications. In other words, existing AppleTalk and TCP/IP software will continue to work.

Open Transport is a general-purpose networking architecture that works now and will work in the future. A future version of Open Transport optimized for Copland will provide further enhancements.

• *QuickTime Conferencing and CyberDog are promising new technologies that may present you with new product opportunities.*

QuickTime Conferencing, which builds on Open Transport, provides a scalable architecture for enabling real-time collaboration across a variety of network connections. Initially, Apple will first make QuickTime Conferencing an option for PCI-based Power Macintosh computers, but Apple plans to make desktop conferencing widely available on the Macintosh platform as soon as possible.

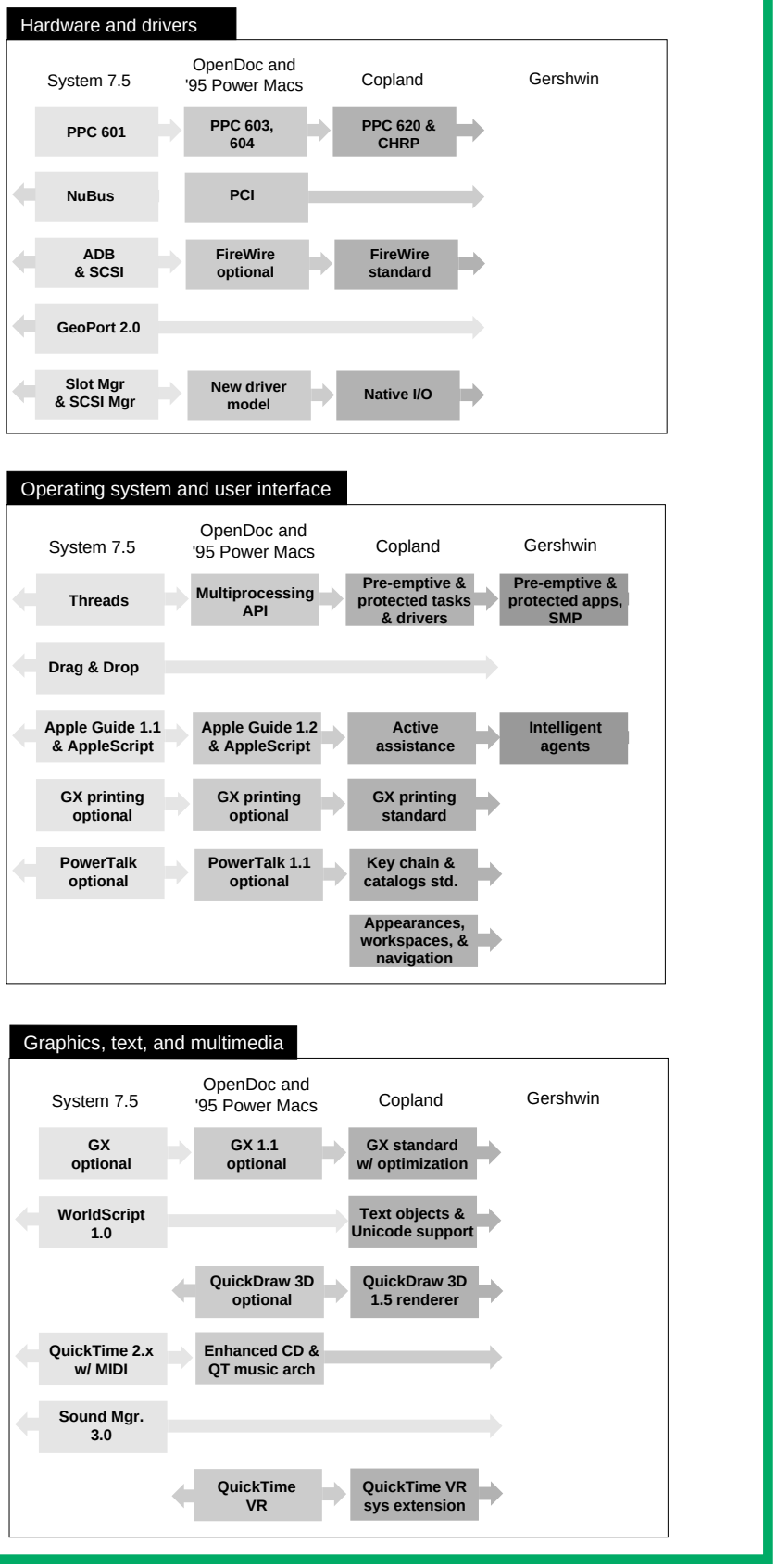
CyberDog is a promising blend of OpenDoc and Internet tools that generated a lot of developer enthusiasm for OpenDoc when it was demonstrated at the WWDC. In the words of one Apple employee, "You could see the light bulbs going on above everybody's heads."

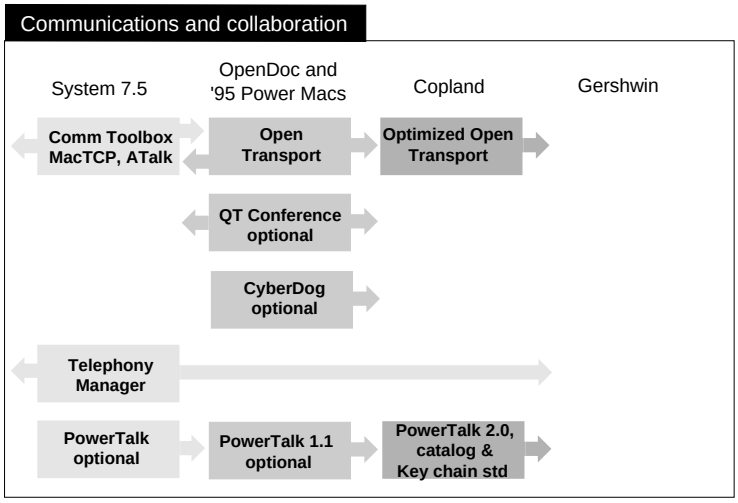
Basically, CyberDog encapsulates Internet access inside OpenDoc parts, vastly simplifying the process of extracting information from the Internet and giving users a simple way to "package" Internet access within a document. Imagine, for example, sending a customer an e-mail message that "contains" your World Wide Web home page. Look for more coverage of the CyberDog technology in *Apple Directions* as it becomes available.

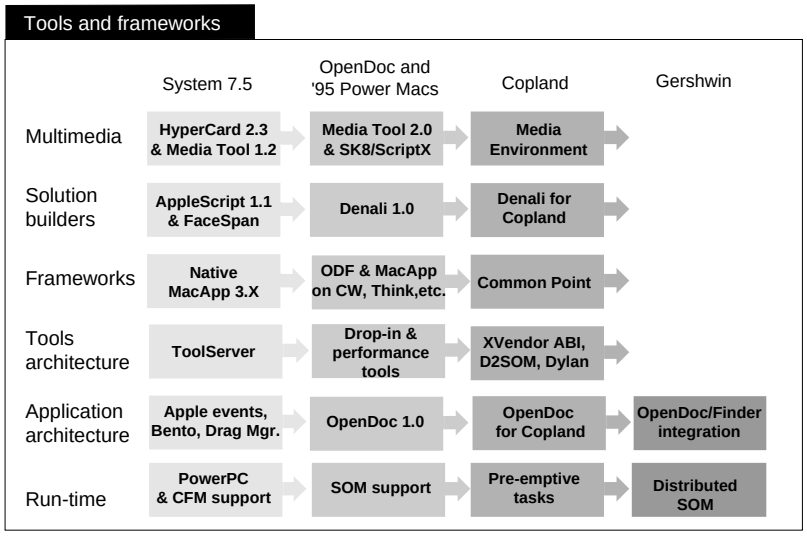
• *The Telephony Manager is the architecture that you will continue to use to integrate the Macintosh with voice, data, fax, and other phone services.*

Just as GeoPort, mentioned earlier, is the hardware connection between Mac OS

The Apple Technology Mosaic







Abbreviations used:

ABI	Application binary interface
ATalk	AppleTalk
CHRP	Common Hardware Reference Platform
D2SOM	Direct-to-SOM compiler
DSOM	Distributed System Object Model
GX	QuickDraw GX
Media Tool 2.0	Apple Media Tool 2.0
ODF	OpenDoc Development Framework
PPC	PowerPC processor
QT Conference	QuickTime Conferencing
QT music arch	QuickTime music architecture
QuickTime VR sys extension	QuickTime VR system extension
SMP	Symmetrical multiprocessing
SOM	System Object Model
Std	Standard
XVendor	Cross-vendor

computers and telephony-based services, the Telephony Manager is the software interface you'll use to create telephony-aware solutions. There are no changes in sight; just use the Telephony Manager, and your telephony-aware application will work with whatever hardware connection your customer chooses.

• *As I mentioned earlier, Apple will continue to improve PowerTalk, which will (by the Copland time frame) provide a standardized "front end" that will unify and simplify users' interactions while they are communicating with the outside world.*

Tools and Frameworks

• *At the run-time level, SOM will provide the foundation for future Macintosh software development.*

The layers stack up like this: On top of the PowerPC processor is the Code Fragment Manager (CFM), which is integral to the Power Macintosh implementation of the Mac OS. System Object Model (SOM), which is a foundation technology for OpenDoc, is built on top of the Code Fragment Manager. Therefore, it's important for development tools and frameworks to support CFM and SOM.

Copland pre-emptive, protected tasks will make development better and easier. Tools will be more efficient and responsive through multitasking. Also, debugging will be less frustrating when development tools are running in a separate task that is protected from the software being debugged.

By or before the Gershwin time frame, the OpenDoc community (including Apple) will extend SOM to provide more software development possibilities by allowing software objects to interact across a network. (Under SOM, only software objects on the same computer can interact.) This extension (formerly known as Distributed SOM, or DSOM) has been incorporated into the SOM architecture to make SOM the one technology that software developers will use, now and in the future, to implement object-based software.

• *At the application-architecture level, OpenDoc is integral to Apple's plans for evolving the architecture of applications.*

OpenDoc 1.0 will ship to customers late this fall—and by the Gershwin time frame, you will see a stronger integration between OpenDoc and the Macintosh human interface.

• *At the tool-architecture level, Apple is committed to driving the evolution of software development tools and frameworks.*

The Mac OS software development landscape already offers ToolServer, which allows development systems from different vendors to communicate and work with MPW tools.

Over the next year, you will see the results of Apple's work with tool developers to create development environments that can be enhanced by drop-in development tools, such as compilers and resource tools. In addition, expect to see a wider range of performance enhancement tools for Mac OS–compatible computers that use the PowerPC processor, as well as enhanced optimizers for Apple's C and C++ compilers.

In the Copland time frame, Apple expects to increase programmer productivity by promoting tools and standards that allow the linking of binary objects created by different programming tools and languages. A Direct-to-SOM facility will enable more efficient and convenient creation of SOM-based software. And, although it is still evolving, the Dylan programming language promises to provide a substantial jump in programmer productivity and program quality.

• *At the framework level, Apple has taken steps to reinvigorate MacApp. Taligent's Common Point framework, which drew a highly favorable response at the WWDC, will be a solution for many in-house developers and consultants in the Copland time frame. The OpenDoc Development Framework will ease the development of cross-platform OpenDoc parts.*

MacApp has been the choice of many developers for many years. To ensure its viability, Apple has ported it to the Power Macintosh platform. Within the next year, Apple will be making additional enhancements to it to support strategic portions of the Mac OS (including making MacApp applications into container applications to support OpenDoc). In addition, development environments—including those of Metrowerks and Symantec—will support MacApp. Because Apple's priority is ensuring that you have the tools you need, MacApp will be included with these development environments at no extra charge.

The OpenDoc Development Framework (ODF, formerly known as OPF) will be available to speed the development of OpenDoc parts and applications for both Mac OS–compatible and Windows-compatible computers. It will evolve as a primary development framework for multiplatform OpenDoc software. (For information on the advantages of using the OpenDoc Development Framework, see "OpenDoc *Is* Cross-Platform," on page 1 of the November 1994 issue of *Apple Directions*.)

• *Today, AppleScript and FaceSpan provide solution builders with a way to create custom solutions from existing components. In the future, Denali will do the same thing in the context of OpenDoc parts.*

Solution builders (usually in-house programmers and consultants) want to develop solutions that meet their clients' needs quickly. Today, that means writing scripts for tying together multiple Apple event–aware applications. You can use FaceSpan™ to build a "dashboard" that serves as the human interface for triggering custom scripts that implement the clients' needed solutions.

Denali is the code name for a jointly developed technology that will allow solution builders to create and integrate OpenDoc parts in a visual way, much like the way Visual Basic customers build and use VBXs to create custom solutions. Its advantage is in its full object orientation and the cross-platform nature of both the Denali environment and the software it is used to create. See the news item on page 10 for more details.

• *HyperCard 2.3 and Apple Media Tool 1.2 give multimedia developers a quick, powerful way to create Power Macintosh native titles. Future technologies, including Apple Media Tool 2.0 and the SK8 authoring environment, will evolve into a tool- and output-independent media authoring environment in the Copland time frame.*

In addition, the Apple Media Tool is a cross-platform tool that allows you to build multimedia titles that run on both Mac OS–compatible and Windows-compatible computers.

SK8 is a development environment for creating multimedia authoring tools for a variety of run-time environments. At the WWDC, Apple demonstrated use of SK8 to create multimedia content in both ScriptX and HTML (HyperText Markup Language) formats. SK8 is a promising technology still under development in Apple's Advanced Technology Group. *Apple Directions* will keep you informed of SK8 developments as they become available.

A Necessarily Complex Strategy

When I look at the five figures that make up this Technology Mosaic, I ask myself, "How do I justify this complexity? Can't Apple make things simpler?" The answer is *no,* but there are some good reasons for it:

• *Apple's strategy encompasses both hardware and software.* If you take away the hardware elements in this Technology Mosaic, the complexity of Apple's strategy decreases by at least 20 percent. However, the fact that Apple's strategy includes both hardware and software is very important—it results in a platform that works better and presents more possibilities because the same hand is making sure that the hardware and software work closely together.

• *Apple's strategy is driven by innovation.* Apple has always been known for its innovation. A statement made in the December 1994 issue of *BYTE* magazine sums it up neatly: "Indeed it would not be an exaggeration to describe the history of the personal computer industry for the past decade as a massive effort to keep up with Apple. . . . [The Macintosh] went on to pioneer or popularize almost every innovation in personal computing."

Here is a partial list of Apple's recent innovations: OpenDoc, the Power Macintosh architecture, GeoPort, Macintosh Drag and Drop, Apple Guide, PowerTalk, Copland human-interface innovations, QuickDraw 3D, QuickTime, QuickTime VR, Open Transport, CyberDog, AppleScript, HyperCard, Apple Media Tool, SK8, and Dylan. And the list is even longer if you include outside technologies that will gain widespread acceptance because of Apple's adoption of them.

Innovation is important because it helps both Apple and you succeed against considerable competition. Innovative technologies make it possible for you to invent completely new uses for personal computers, thus giving you a strong market for your products and increasing the strength of the Mac OS platform. Innovation also attracts the most intelligent and creative people—both to develop products for the Mac OS platform and to be the customers (and promoters) of it. Finally, by using innovation to "change the rules" of the personal-computer industry game, Apple can turn the table on its competition by being in control of industry standards instead of having to accommodate standards set by others.

• *Apple's strategy is complicated by the desire to reinvent the Mac OS platform without abandoning today's customers and developers.* Sure, if Apple were starting from scratch, this Technology Mosaic would be a lot simpler. But Apple doesn't have that option. It has to make a transition between what exists now and what it wants to exist, eventually—and transitions require extra steps.

By the time the Mac OS reaches the level of Gershwin, it will have been completely redesigned from top to bottom. The Mac OS will be a powerful new platform running on RISC and poised for the next decade's growth. (We doubt that the competition will be able to say the same.) And Apple expects to gain, not lose, customers along the way—not a small accomplishment!

Apple's strategy is complex because it is trying to accomplish a lot. But the bottom line is that you should study the entire strategy, then focus your attention on those parts that are relevant to what your company is trying to do.

As always, let me know what you think; I'll forward your feedback to Apple management. And be sure to read next month's column, which will complete the process of describing Apple's technology strategy. ♣

Apple News

New Power Macintosh

continued from page 1

• Quadruple-speed internal CD-ROM drive

Another model, the Power Macintosh 9500/132, retails for $5,799 in the United States. It differs from the Power Macintosh 9500/120 in the following ways:

• Its PowerPC 604 processor runs at 132 MHz.

• It has a 2-GB internal hard disk drive.

• It does not include a graphics card (to allow users to buy the high-end video they want without paying for hardware they won't use).

PowerPC 604 Processor

The PowerPC 604 processor is the first second-generation PowerPC processor to be used in an Apple product. Reports indicate that a given PowerPC 604 processor offers up to 1.5 times the performance of a PowerPC 601 processor running at the same speed. Apple expects the Power Macintosh 9500 to perform up to six times faster than a Macintosh Quadra 950 and up to two times faster than the Power Macintosh 8100/110, which was previously the fastest personal computer from Apple Computer, Inc.

One welcome feature of the Power Macintosh 9500 computers is that the processor and clock chip are on a detachable processor card. In the future, Power Macintosh 9500 customers can preserve their investments in their computers by replacing the processor card with a faster version.

PCI Bus

The Peripheral Component Interconnect bus is an industry-standard expansion bus that offers customers significantly higher performance than existing NuBus solutions. With the addition of the appropriate software drivers, PCI cards that comply with the PCI 2.0 specification are expected to work in the Power Macintosh 9500 and to perform up to three times faster than similar cards on even the fastest NuBus systems. The adoption of PCI is part of Apple's commitment to provide "fit-in" compatibility with industry standards.

The move from NuBus to PCI not only allows PC hardware developers to sell their products to the Macintosh market, but traditional NuBus developers can now leverage their solutions into the 80x86-computer marketplace with little additional investment.

Because the second-generation Power Macintosh architecture was designed around a PCI-only strategy, PCI cards in the Power Macintosh 9500 run at the cards' maximum speed of 33 MHz. (This contrasts with many 80x86-based computers that have a mixed bus and cannot run their PCI cards at full speed.) In addition, PCI is a scalable technology that will offer improved performance in the future; Apple is committed to using the PCI bus in the long term and to upgrading PCI as its technology improves.

For the Macintosh community, Apple's adoption of the PCI bus means customers will soon have access to a much wider selection of computing, enterprise computing, data acquisition, and networking solutions than is available

The Power Macintosh 9500.

for NuBus-based Macintosh computers—and at more affordable prices. Thirty-five companies have announced their support for Power Macintosh and PCI, including such leading PCI vendors as ATI Technologies, Diamond Multimedia Systems, and Matrox Graphics.

Open Transport

The Power Macintosh 9500 features Open Transport, which is Apple's new networking architecture that will allow the same communications application to work with the customer's choice of networking protocol. (Existing communications applications will continue to run correctly on Power Macintosh 9500 computers.)

Open Transport gives developers a larger market to sell to, makes it easier for network administrators to do their jobs, and gives customers a more powerful and intuitive human interface for dealing with their networks. (Look for an article on Open Transport in an upcoming issue of *Apple Directions.*)

Memory

Though the Power Macintosh 9500 computers come standard with 16 MB of memory, they can be expanded to a maximum of 768 MB—that is, three-quarters of a *gigabyte*—of memory. To do this, the computers use a new 64-bit, 168-pin dual-inline memory module, or DIMM.

High Capacities

Speed, maximum memory, and high-capacity hard disks make Power Macintosh 9500 computers ideal for high-end uses that include color production, prepress work, multimedia development, video editing, technical document production, and CAD (computer-aided design). In addition, the computers' six PCI slots make them attractive to scientific users who need multiple card slots for their data acquisition equipment.

System Software Improvements

The Power Macintosh 9500 ships with the latest version of the Mac OS—Macintosh System 7.5.2—which includes several new performance enhancements and features that have been optimized to take advantage of the new Power Macintosh architecture and PowerPC 604 processor. In particular, Macintosh System 7.5.2 includes increased support for the built-in AV (audio-visual) technologies, including improved control panels to make these technologies easier to use.

The Power Macintosh 9500 computers also feature a new 680x0 emulator that increases 680x0 software performance by up to 25 percent. The new emulator, in conjunction with the PowerPC 604 processor, is expected to yield the fastest performance of 680x0 applications of any Macintosh computer.

Other Features

In addition to the standard Apple back-panel connectors (including SCSI, Ethernet, two GeoPort-compatible serial ports, and 16-bit stereo sound input and output), the Power Macintosh 9500 computers include an Apple SuperDrive floppy disk drive and an internal, quadruple-speed AppleCD 600i CD-ROM drive. The Power Macintosh 9500 case also includes room for additional mass storage—one 3.5-inch half-height bay for removable media and two 3.5-inch half-height bays for internal hard disk drives.

Denali: OpenDoc-Based Tool for Custom Solutions

At the Worldwide Developers Conference last month, Apple Computer, Inc., and IBM Corp. announced their intent to provide multiplatform application development tools, which will enable users to build custom applications using OpenDoc technology.

These tools, code named *Denali,* will provide in-house developers as well as consultants, value-added resellers, independent software vendors, systems integrators, and other solution providers the ability to create and script OpenDoc components to build custom applications. These tools will bring BASIC visual programming to the Mac OS, OS/2, AIX, and Windows platforms.

This newly announced development environment is expected to do the following:

- run on Mac OS, OS/2, AIX, and Windows operating systems
- incorporate BASIC language technology, similar to Microsoft Visual Basic, to build and interconnect OpenDoc parts
- include object-oriented extensions to the BASIC programming language
- provide the ability to create OpenDoc parts and script them into composite applications
- exploit the IBM System Object Model (SOM), a language-neutral, object-structured protocol that allows programmers to write object-oriented applications regardless of what programming language was used to create the object
- provide access to many data sources through multiple data access protocols, including the open database connectivity (ODBC) services

The parties involved said that they were announcing Denali as a technology, not a product, but that developers would probably see usable tools within a year.

"This initiative reinforces IBM's and Apple's commitment to provide advanced, multiplatform programming tools that enable users to extend existing BASIC applications through the use of OpenDoc technology," said Tim Negris, vice president, Application Development Marketing, IBM Software Solutions Division. "Our combined efforts will allow developers to build new, advanced, native OpenDoc solutions as well as redeploy existing BASIC applications to a significantly broader user base on OS/2, AIX, Mac OS, and Windows platforms."

Ike Nassi, vice president, Apple System Software Technology, added, "This effort represents an important step in providing powerful tools that will support OpenDoc and provide a compelling Visual Basic–style solution to Macintosh users. These custom application development tools support the move to RISC-based PowerPC development and meet customers' growing demands for multiplatform development capabilities."

"This announcement is of strategic importance to both developers and users," said Michael Barton, vice president of marketing for Component Integration Laboratories (CI Labs), the open, vendor-neutral association founded to drive the adoption of component software through the OpenDoc standard. "The tools will provide developers with a powerful visual development environment that is multiplatform and built on component software standards. Users will also benefit through intuitive, cross-platform solutions that share a common user interface."

Apple Makes QuickTime VR, Apple Media Kit More Affordable

Apple Computer, Inc., recently announced three developments that should make QuickTime VR and the Apple Media Kit more attractive to developers of multimedia products.

First, Apple has reduced the QuickTime VR licensing fee for commercial use to between 3¢ and 8¢ per unit (down from as much as 80¢ per unit) and dropped the licensing fee for noncommercial use.

Second, Apple has reduced the cost of acquiring the QuickTime VR development software to as low as $495, down from $1,995.

Third, Apple has dropped the licensing fee for the run-time version of its Apple Media Kit multimedia authoring software; also, there is no licensing fee for QuickTime or QuickTime for Windows.

QuickTime VR Authoring Tools Suite

QuickTime VR—short for *virtual reality*—is an Apple technology for Mac OS–compatible and Windows-compatible computers that lets users view a photographic or rendered representation of a 360° scene and feel as if they were actually at the scene. In addition, users can pick up and interact with objects at the scene.

The QuickTime VR Authoring Tools Suite includes the QuickTime VR run-time software (which allows consumers using ordinary personal computers to interact with QuickTime VR scenes and objects) and QuickTime VR Authoring Tools, which customers can use to build interactive 360° views, link them together, add hot spots, and create interactive objects.

The QuickTime VR Authoring Tools Suite will now be offered in two configurations:

- the suite alone, available in the United States for $495
- a bundle that includes the QuickTime VR Authoring Tools Suite, the MPW Pro development environment, and one seat at a QuickTime VR training course offered by Apple's Developer University—priced at $1,995.

QuickTime VR Authoring Tools Suite version 1.0 will initially ship with final versions of the authoring tools and the Macintosh run-time software, and a beta version of the Windows run-time software. The final version of the Windows run-time software will follow as a free upgrade. Version 1.0 of the Windows 3.1 run-time software is expected to be available in June 1995.

To order the QuickTime VR Authoring Tools Suite 1.0, call APDA, Apple's source for developer tools, at 800-282-2732 (U.S.), 800-637-0029 (Canada), or 716-871-6555 (international).

QuickTime VR Licensing and Royalties

The QuickTime VR run-time software can be distributed by developers along with their QuickTime VR commercial CD-ROM titles for the following per-unit royalties:

- fewer than 25,000 units—no charge
- over 25,000 units—$400 for each set of 5,000 units in excess of 25,000 units

In addition, a new Enhanced CD format allows the creation of audio CDs that play music normally in a CD player but provide extra functions when placed in a personal computer's CD-ROM drive. For developers who want to create Enhanced CDs that use QuickTime VR to augment their audio content, the following per-unit royalties apply:

- fewer than 50,000 units—no charge
- over 50,000 units—$750 for each set of 25,000 units in excess of 50,000 units

This QuickTime VR royalty structure—less than 8¢ per CD for software titles and 3¢ per CD for audio titles—represents a significant reduction from the original per-unit cost, which ranged from 40¢ to 80¢. There are no royalties charged for noncommercial use of QuickTime VR.

Developers using QuickTime VR can create content on the Macintosh platform and deliver it to run on both Mac OS–based and Windows-based personal computers, enabling access to the vast market of personal computer users.

QuickTime VR Training and Support

The QuickTime VR Authoring Tools Suite is designed for use by experienced multidisciplinary teams, including professional photographers and multimedia developers. In the future, Apple expects to develop a simplified version of the QuickTime VR Authoring Tools Suite for a broad range of users interested in developing multimedia and virtual-reality titles.

In response to developer demand for QuickTime VR training, Apple Developer University has begun offering a three-day "Multimedia Development With QuickTime VR" class. The course, suitable for multimedia designers and title developers, will cover all the steps involved in planning, creating, and developing a QuickTime VR scene. The class will cost $1,500.

Apple will provide QuickTime VR technical support for members of the Macintosh Partners developer program. For more information about the QuickTime VR class dates and training, or for information about Apple Developer Programs, call the Apple Developer Hotline at 408-974-4897.

QuickTime VR System Requirements

The QuickTime VR Authoring Tools Suite 1.0 requires a minimum of a 33-MHz 68040-based Apple Macintosh computer with 40 MB of memory, System 7.1, MPW 3.2 or greater, and HyperCard 2.2. The run-time software requires HyperCard 2.0 (or later) or Macromedia Director 3.1.3 (or later).

The minimum system configuration required for the run-time software is a QuickTime-capable Macintosh with a 25-MHz 68030 processor, QuickTime 2.0, System 7.1 or later, 8-bit video, and 8 MB of memory. A double-speed CD-ROM drive is recommended for CD-based software titles.

The minimum Windows system required for the run-time software is QuickTime 2.0 for Windows, Windows 3.1, and a MPC 2 configuration (33-MHz 80386 processor, 8-bit video, 8 MB of memory). A double-speed CD-ROM drive is recommended for CD-based software titles.

Apple Media Kit Licensing and Royalties

The Apple Media Kit—which consists of the Apple Media Tool for authors, designers, and educators, and the Apple Media Tool Programming Environment for programmers—is Apple's cross-platform multimedia authoring tool. With one development effort, developers can create multimedia titles—including interactive multimedia catalogs, rock music titles, online advertising, and sophisticated networked kiosks—for both Mac

OS–compatible and Windows-compatible computers.

Effective immediately, the Apple Media Kit run-time software, as well as QuickTime and QuickTime for Windows, can be distributed at no charge. The run-time software is the software that allows a title created with the Apple Media Kit to be played back without the authoring tool.

The Apple Media Tool is available through all Apple software distribution channels, including APDA, dealers, and retail software outlets. In the United States, pricing is as follows: Apple Media Tool, $599; Apple Media Tool Programming Environment, $995; Apple Media Kit, $1,195. Special higher education pricing and an advanced support option are also available.

Apple's European Newton Developer Programs

Apple has begun to make Newton developer support programs available throughout Europe. Newton developers in France, the United Kingdom, Germany, and Spain, as well as Belgium, the Netherlands, and Luxembourg (Benelux) now have access to Newton Associates programs. Apple also expects to introduce Newton Associates Plus and Partner programs in most of those countries by the middle of the year.

All European Newton Associates receive the Newton Developer Mailing (including the Developer CD, the *Newton Technology Journal,* and *Apple Directions*), access to technical information on AppleLink, discounts on hardware, software, and training, and marketing support. Each country's program also offers additional services customized for the developers in that locale.

If you're a Newton developer in France, the United Kingdom, Germany, Austria, Spain, or one of the Benelux nations, you can get more information about the Newton Associates program in your country by contacting your local Apple office. Phone numbers and AppleLink addresses for Apple offices in those countries are as follows:

- *France:* Telephone 05-90-72-26; AppleLink FRA.DEV.PROG
- *United Kingdom:* 0800-50-50-94 or 44-131-458-6736; UK.DEV.PROG
- *Germany and Austria:* 49-89-99640-534; CE.DEV.ADMIN
- *Spain:* 07-34-1-663-17-80; SPA.TPS
- *Benelux:* 31-3405-93900; BNL.DEV

Developers in other countries may purchase the Newton Developer Mailing from APDA. For APDA ordering information, see page 36.

Apple Announces Sweeping Update to Printer Product Line

Apple Computer, Inc., recently introduced three additions to its printer line and significantly updated one of its existing printers. Apple now offers two color ink-jet printers with high-end usability features usually found only in more expensive printers, a monochrome laser printer that breaks a price/performance barrier for high-quality desktop laser printing, and a color printer that provides professional-quality output usually performed by imaging devices that are as much as ten times more expensive than Apple's new offering.

On June 19, Apple announced the following:

- the Color StyleWriter 2200, a portable color ink-jet printer that's small enough to fit in a standard briefcase
- changes to the Color StyleWriter 2400, including a variety of new features and a price reduction
- the LaserWriter 4/600 PS, which offers 600 dots-per-inch (dpi) PostScript™ printing for under $1,000 (U.S. price)
- the Color LaserWriter 12/600 PS, a top-quality true 600-dpi color printer for both Mac OS and Windows systems that provides near photographic imaging and retails for under $7,000 (U.S. price)

Details about three of the four products follow. *[Editor's note: We'll tell you more about the Color LaserWriter 12/600 PS in next month's issue; it introduces a new print architecture and a variety of new features that we want to spend a little more time explaining.]*

Color StyleWriter 2200

The new Color StyleWriter 2200 is a truly portable color ink-jet printer for mobile professionals. It delivers the same laser-quality printing of text and graphics provided by other printers in Apple's color ink-jet line, yet weighs just over three pounds and measures less than 2 inches across and 2.2 inches high. It introduces the following features:

- Desktop printing, which lets users drag and drop files onto a printer icon on the desktop. This action automatically launches the application and the print dialog box.
- Two-up and four-up printing, which let users print two or four reduced-sized documents on a single page, saving money and paper.
- the watermark feature, which lets users place words like *draft* or *confidential* on the background of a page. The Color StyleWriter 2200 includes eight watermarks, but any PICT or PICT 2 file (such as a color company logo) can be used as a watermark.

The Color StyleWriter 2200 printer ships with a color cartridge that offers 360-by-360 dpi color and true black printing on the same page. It uses water-resistant ink and can print a color page in under three minutes. A high-performance black ink cartridge provides 720-by-360 dpi edge-smoothing technology for crisp text output in "best" mode, or up to five pages per minute laser-fast output in "normal" mode. A 30-sheet paper feeder is built in, and the printer easily handles plain, coated, or glossy paper, transparencies, back-print film, and envelopes. The printer offers 64 TrueType fonts for high-quality text, matching the font set included with Apple LaserWriter printers.

The Color StyleWriter 2200 is compatible with all 68020 or faster Macintosh computers, and comes with a PowerPC "native" driver that greatly enhances performance when printing complex color documents from Power Macintosh computers. The QuickDraw GX driver gives users access to Apple's latest imaging architecture. The printer also supports ColorSync 2.0, an advanced technology that provides the closest possible match between colors viewed on a display and the color that comes out of the printer.

An optional nickel-metal hydride battery is available for use with the Color StyleWriter 2200. With the battery installed, the Color StyleWriter 2200 weighs only 4.2 pounds and can print

approximately 200 pages between charges. An optional universal AC adapter for use with all standard power outlets is also available.

In the United States, the Color StyleWriter 2200 has an Apple price of $419. Customers outside the United States should contact their local Apple representatives for information about pricing and availability.

Updates to Color StyleWriter 2400

The Color StyleWriter 2400 printer has been updated to include features similar to those offered by its portable sibling, the Color StyleWriter 2200, at a similar price, yet in a form factor more suitable for nonmobile users in home, education, and small business environments.

It also supports ColorSync 2.0 and QuickDraw GX and provides desktop printing, two-up and four-up printing, and the watermark feature. The Color StyleWriter 2400 has been enhanced to offer 720-by-360 dpi true black printing at up to five pages per minute. Color printing is accomplished at 360-by-360 dpi at a rate of three minutes per page with a water-resistant color ink cartridge. It ships with Apple ColorShare technology, which lets multiple users share the printer over a variety of networks. Its built-in paper tray holds 100 sheets or 15 envelopes and easily handles a variety of media. The printer offers 64 TrueType fonts, a PowerPC native driver, and a built-in power supply.

The enhanced Color StyleWriter 2400 is available worldwide; it has a U.S. Apple price of $429.

LaserWriter 4/600 PS

With the newest member of the LaserWriter family, Apple has made PostScript printing more affordable than ever before for home, education, and small business users. The LaserWriter 4/600 PS, which weighs only 15 pounds, is capable of printing 106 line screens and 122 shades of gray for top grayscale imaging. As the "4/600" part of its name implies, it prints at a rate of four pages per minute with 600-dpi resolution. It also includes Apple's new desktop printing feature.

The LaserWriter 4/600 PS ships with 64 TrueType and PostScript fonts, features a speedy RISC processor, and includes built-in LocalTalk connectivity. It's also equipped with the LaserWriter Bridge 2.0 software, making it capable of integrating into EtherTalk and TokenTalk networks. It complies with EPA Energy Star standards and features an energy-saving sleep mode that completely shuts the printer down between uses. "Instant awake" technology all but eliminates printer warm-up time. When it's not in use, the printer runs on less than 10 watts of power and makes virtually no noise.

The Apple LaserWriter 4/600 PS is available worldwide and has a U.S. Apple price of $929.

QuickDraw 3D Beta Ships

Just in case you missed it, last month's installment in the Developer CD (June 1995; Reference Library Edition) contained the beta version of QuickDraw 3D, an application programming interface you can use to create and manipulate electronic three-dimensional graphics. In addition, Apple published the specification for the platform-independent 3D metafile format (3DMF) promised as part of the original QuickDraw 3D announcement; the specification establishes a standard 3D data type that all applications can read and write.

Many of you have been enthusiastic about QuickDraw 3D. More than 30 developers from around the world have already announced their support of the new technology. John Wilczak, chairman and CEO of HSC Software, the developer of Kai's PowerTools among other products, says, "QuickDraw 3D is . . . destined to lead the industry, just as QuickTime has, by making it easy for developers to rapidly integrate 3D into applications. The winners are artists and creatives who use the Power Mac."

Kevin MacGillivray, vice president of Radius's Color Publishing Division, says that "Radius is excited about the opportunity for workstation-level 3D graphics and performance on the desktop. Apple's QuickDraw 3D is a major enabling software technology that, when boosted with 3D hardware, will revolutionize the creation of content. Soon customers in color publishing and broadcast design will be able to produce 3D illustrations, product designs and animation with the quality and speed they require, on their platform of choice."

QuickDraw 3D consists of human interface guidelines, a high-level modeling tool kit, a "plug in" shading and rendering architecture, a device and acceleration manager for hardware acceleration and the 3DMF platform-independent file format. Each part of the API is extensible, making QuickDraw 3D one of the most open graphics architectures in the industry. All QuickDraw 3D components are designed to provide you and your customers with the flexibility to customize and extend 3D applications for user- specific needs. QuickDraw 3D requires a Power Macintosh system running Version 7.1.2 or later of the MacOS, a hard-disk drive, and 16 MB of memory.

3DMF is a cross-platform file format that supports 3D data, not only geometric information typically used in 3D file formats but also appearance information such as textures, lighting, and shading. 3DMF also includes the ability to preserve custom information such as universal resource locators (URLs). Because Apple designed 3DMF as a platform-independent standard, files saved in the 3DMF format preserve all geometric and appearance data regardless of the application or operating system used to view 3D files. Regardless of the platform you develop for, you can adopt 3DMF as a means to move 3D data between applications or over the Internet.

Apple is currently providing you with the following information: 3DMF file specification in a FileMaker Pro database and PDF format; the QuickDraw 3D Parser for Windows, Unix, and 680x0 Macintosh systems that allows you to read and write 3DMF data from any application on any platform; samples of 3DMF models; and a custom registry where developers can register with Apple their unique extensions to QuickDraw 3D. You can find this material, along with the QuickDraw 3D beta release, on the June 1995 Reference Library Edition of the Developer CD. You can also find the above information on the Internet at Apple's home page (http://www.info.apple.com).

Final QuickDraw 3D software is expected to be available this summer as part of a comprehensive software developer's kit. Also beginning this summer, Apple plans to equip its Power Macintosh computers with QuickDraw 3D capabilities. The software is expected to be incorporated into Copland, the next major release of the Mac OS. A Windows version of QuickDraw 3D is expected early next year. ♣

Technology

CD Highlights

System Software Edition, July 1995

Once again, it's Worldwide System Software time!

This installment in the monthly Developer CD series contains U.S. and Worldwide versions of the following software:

- System Software 7.5
- System 7.5 Update 1.0
- PowerPC 601 Processor Upgrade Enabler
- System Enabler 406 for the Power Macintosh 5200 and 6200 computers
- PowerPC Enabler for the second generation of the Power Macintosh 6100, 7100, and 8100 computers

In the System Extensions folder, you'll find an assortment of versions of QuickDraw GX 1.1.2 for the world's different script systems.

So, in addition to the latest exciting chapter in the Gestalt Selectors List saga, here are the new and updated items on this month's disc.

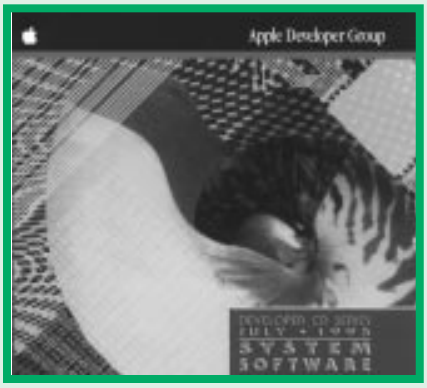
Copland WWDC Materials

The two documents here are required reading for anyone interested in the future direction of the Mac OS. *Copland Technical Overview* is a detailed overview (about 150 pages) of the Copland OS architecture. *The Microkernel White Paper* describes Apple's plans for the microkernel to be used in future operating systems. Apple is soliciting your thoughts on Copland and the microkernel; you can address feedback to AppleLink address COPLAND or Internet address copland@applelink.apple.com.

Designing PCI Cards & Drivers

This book describes the Macintosh implementation of the Peripheral Component Interconnect (PCI) local bus established by the PCI Special Interest Group. The PCI local bus standard defines a high-performance interconnection method between plug-in expansion cards, integrated I/O controller chips, and a computer's main processing and memory system.

The first generation of Power Macintosh computers—the Power Macintosh 6100, 7100, and 8100 models—supported NuBus expansion cards. Subsequent Power Macintosh models, including the just-released Power Macintosh 9500 series, support the PCI standard. (See the news story on page 1 of this issue for coverage of the Power Macintosh 9500 announcement.) This book contains useful information for product developers who want to design PCI expansion cards (and their associated software) that are compatible with the new computers.



System Software Edition

The information in this book provides general information about PCI cards and drivers. You should also refer to the developer notes that accompany each Macintosh product release for exact details about how that product implements PCI.

Inside This Section

Developer Notes Update 7/95

Included here are developer notes for Apple's latest hardware products, the Color LaserWriter 12/600 PS (see news on page 12) and the Power Macintosh 9500 series of computers.

Disk I/O Performance Tools Update

This update corrects a minor bug in the IOTracer Analyzer application distributed in the Disk I/O Performance Tools package on the June 1995 Developer CD. It will be rolled into the complete package in the next Tool Chest CD, which will be published in August 1995.

DTS QT Utilities.May-95

This is a collection of useful QuickTime functions and applications. Consult the documents in the Documentation folder for more details. For information on updates and bug fixes, see the document What's New, May-95.

International NSI 1.5.1

This folder provides an update of networking software for the Macintosh computer. See the Read Me document for details.

Japanese Language Kit Version 1.2

The Japanese Language Kit is a combination of system and language resources that enables

please turn to page 27

OpenDoc Human Interface: Frequently Asked Questions From the WWDC

By Dave Curbow, Elizabeth Dykstra-Erickson, Kerry Ortega, and Geoff Schuller, OpenDoc Human Interface Team

Here's a collection of questions the OpenDoc Human Interface team was asked during the recent Worldwide Developers Conference (WWDC). We hope you'll find our answers helpful.

Q: Where can I find a copy of the **OpenDoc Human Interface Guidelines**?
A: They are on the *1995 WWDC Technologies* CD that was distributed at the conference. To find the guidelines file, use the path 1995 WWDC Technologies:OpenDoc:Documentation:Human Interface:OpenDoc HI Guidelines. (To read the file, you'll need to install Adobe™ Acrobat Reader, which you can find on AppleLink.) We welcome your comments on these guidelines.

[Editor's note: The 1995 WWDC Technologies *CD is also included in this month's Apple Developer Mailing.]*

These guidelines are up-to-date, with one exception: The section "Feedback for Cropped Parts" on pages 101–103 should have been deleted.

Q: Why doesn't OpenDoc display the "not" symbol (circle with a slash) when a user is dragging objects over invalid drop targets? Displaying this symbol would be very helpful.
A: According to the *Macintosh Drag & Drop Guidelines,* feedback shows users where they may drop, rather than where they may not drop. This is consistent with our general philosophy of using visual feedback to show users what actions are available at any given moment. However, other platforms (for example, Windows and OS/2) use an "invalid drop" feedback mechanism such as the one you describe. On those platforms, OpenDoc follows those user interface guidelines.

Q: Why can't I select the root part in a document? I want to apply some operations to the entire document.
A: Selection applies to content. At the root level of an OpenDoc document, users can select all of the content, but they can't select the container of the content—the document itself. To do that, they have to go to the Finder. (To help you understand what the root of an OpenDoc document is, consider a text document that has a spreadsheet part embedded in it. In this case, the document's root part is a text part.)

Today, if users wish to modify the entire document, they can do so only to the extent that the application provides document-wide commands. OpenDoc is just the same, in that the ability to modify an entire OpenDoc document must rest in the part editor for the root part of that document.

But OpenDoc is different from today's applications in that embedded parts can be turned into separate documents, and separate documents can be embedded as parts inside a container document. This is a powerful way of composing content, but it means that your parts may have somewhat different behaviors when they are embedded parts than when they are the root part.

In particular, when your part is a root part, you may provide additional Document menu commands or capabilities such as window magnification. This is how you must implement document-wide commands, since users cannot select the entire document. In sum, when a part is the root part, it may provide a different set of capabilities or commands than when it is an embedded part.

Some commands, such as Save or Document Info, apply *only* to the document as a whole; and some operations you definitely would not want to operate on the root part, such as Cut. Because you cannot select the root part, this situation cannot arise. Some Document commands, such as Print and Close, apply to the active window. Save a Copy applies to the active draft when a previous draft is frontmost. *[Editor's note: A* draft *is a "snapshot" or a saved state of the document at a given moment. OpenDoc allows the user to view any draft of a document.]*

Q: How does OpenDoc handle saving the content of all of the parts in a document when one of the part editors that is being used in the document crashes? For instance, suppose I have two different types of content (which use two different editors) in a document. I edit the first of these; then, while I'm editing the second part, the first editor crashes. What happens to the changes that I put into the first part? Are they saved or lost?
A: The behavior is the same as it is in today's application environment—if there's a crash, all unsaved changes are lost. Parts do not save independently; all of the parts in the document are saved in one file at once.

Q: How does keyboard navigation work in a compound document? What Command key equivalents are used, and in what order do they navigate the hierarchy of embedded parts within a document? This is an issue for users who prefer to (or absolutely must) use the keyboard to navigate through their documents.
A: This is a complex issue in OpenDoc because there is more than one sort of navigation. Navigation means moving the insertion point or selection within a single part, between a part and its container, or between a part and an embedded part. The arrow keys are typically used to move the insertion point within a single kind of content, as in today's applications.

We haven't yet defined how to move between a container and one of its embedded parts. We are leaning toward using Command– Option–Up Arrow and Command–Option–Down Arrow for moving the insertion point up and down the part hierarchy. That is, when an embedded part is active and the user types Command–Option–Up Arrow, the parent part becomes active and the previously active part is selected. Likewise, when a part is selected, Command– Option–Down Arrow activates that part and places an insertion point as appropriate to that part. Tab might also be used to move among embedded parts. (Note, however, that there is some interaction between Tab and arrow key combinations in this context that we haven't yet defined.) We

please turn to page 18

Human Interface

Do the Right Thing

By Peter Bickford

Dear Doc:

I loved the "Fad Gadgets" article. I've been making fun of the Word toolbars for years and I'm glad to see that somebody agrees with me. Word exacerbates the problem of screen clutter by making the user go to separate menus to shut off the Ruler, Ribbon, Toolbar and Status Bar, essentially guaranteeing that one of the useless things will be left around.

Have you considered the interface error of giving a user too much freedom to control the look and feel of his application? Microsoft Word, for example, lets any command go on any menu. The System 7 Finder allows any icon to be given to any file, application, or folder. This is bad. It would be much better to give the user the ability to arrange icons and menus within limits, so that the graphical grammar of the application remains intact.

More specifically, applications should always have that little operative hand; folders should always have a folder in them. File management commands differ from text-editing commands, and the position of these commands on menus should reflect that. Otherwise, the machine becomes, in time, illogical and difficult to use, especially for anyone but the person who customized the interface.

This problem may become much worse with OpenDoc.

—Michael J. Stern

Dear Michael,

Thanks for the great letter! In 231 short words you've managed to make what may be one of the most difficult points in human interface design. I'll try to elaborate on your points, while at the same time

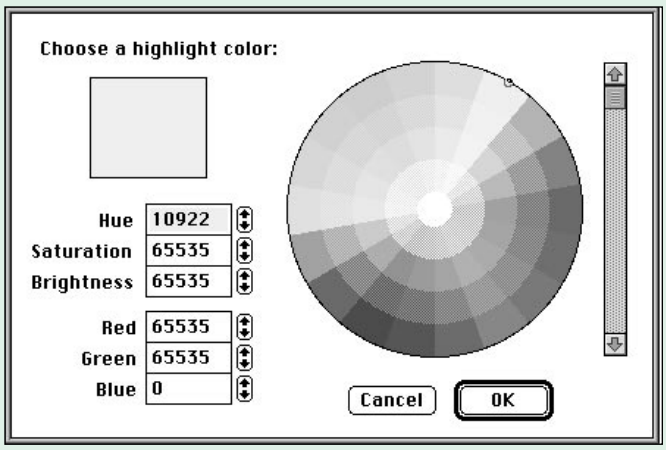
- giving tips on using "hard" and "soft" constraints to do social engineering on users
- confessing one of my darkest fears for the future of the interface
- talking about how to handle consistency when the Macintosh interface suddenly has the ability to match the appearance of your stereo gear

The Paradox of Constraints

One of my first articles for *Apple Directions* talked about the use of constraints. The idea was that by limiting users' freedom (for instance, by dimming buttons that aren't usable in a given situation), you can often make it easier for users to get work done.

The paradox of constraints is that you are helping users get what they really want by taking away some of their options. One of the examples I gave was of a Japanese vending machine that would light the selection buttons of only those products that could be purchased with the amount of money you had inserted. I suppose it could have left all the buttons lit, but pressing them would only have resulted in an error message. Instead, the machine gently guided users by presenting them with only the valid options. In doing so, the machine (which had dozens of products and selection buttons) became much less complex to use.

A "softer" type of constraint is found when users choose a highlight color for text on the Macintosh computer. In the old days of the Macintosh II, the system would simply present users with a color wheel and have them select the exact hue from the entire RGB color space:

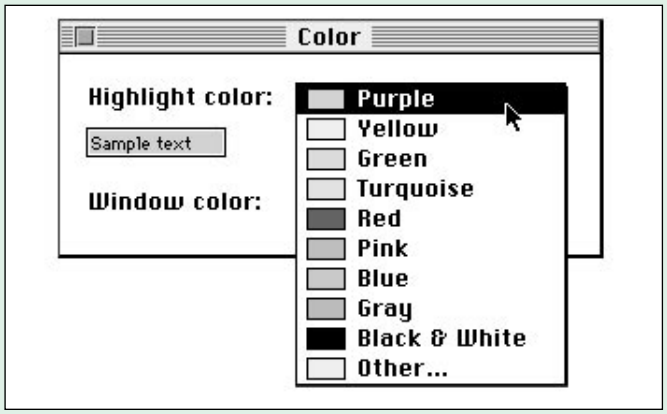


While it may have seemed great that users were being given so many options, in reality, this interface made choosing an appropriate color more difficult than it should have been. For instance, there are a few thousand shades of brown, gray, and dark blue that make really terrible highlight colors—users would end up using a dark shade to select black text. Extremely light colors are also unusable, since the highlighting itself would not be visible against a white background.

Today, users are asked to choose their highlight color from a pop-up list like the one on the next page.

There are two important things to note here. The first is that users are presented with a list of only ten options, instead of the theoretical maximum of 65536 x 65536 x 65536. None of the options result in unreadable text—users are more likely to to get a useful highlight color using the ten options than they were with millions.

Second, notice that users still have the ability to choose "Other," which displays the color wheel and lets users choose any conceiv-



able color. This design hasn't entirely taken away the freedom to choose really bad highlight colors, but it has made that option slightly more difficult to get to. As a result, the user is more likely to choose from one of the "standard" colors, and thus be guaranteed a usable highlight color.

Make "Doing the Right Thing" the Path of Least Resistance

In the best of all possible worlds, people would always make "good" choices and we'd never have crime, war, or application icons that look like folders. When we find that we do need to steer a person's choices, we can go about it in two ways. First, we can take away the choice entirely—we can set up "hard" constraints, like locked doors and dimmed buttons that ensure that folks are never given the opportunity to choose something we don't want them to.

Often, however, it's possible to use softer constraints (such as the pop-up color menu), which simply make it easier to do the right thing than it is to do the wrong thing. This is really a much subtler way to guide the user, since it simply makes "doing the right thing" the path of least resistance. At the same time, it allows for the possibility that users may have special needs that we haven't anticipated and lets them choose those options if they really want or need to. In the highlight color example, for instance, users may be using an inverted screen for CAD drawings, and may therefore need a very dark highlight color to select the light label text. Or it could be that they simply have a fascination with fuchsia and will never rest until they can use it to highlight their text.

In the icon example you mentioned (yes, I'm actually working my way around to addressing your issues, Michael!), Apple probably should have provided some sort of standard templates for folders, applications, and files. It would be great, for instance, if system software automatically "knew" that the Star Trek insignia icon you are pasting onto a folder's icon should be shrunk and embedded onto a standard folder icon—instead of, as is currently the case, the Star Trek icon being visually substituted for the folder icon. (In fact, there's a freeware program that does just that—it's written by Gregory Robbins and is called Folder Icon Maker.)

If Apple had been able to implement such a scheme for custom icons, the "easy way" for users would have resulted in icons that are distinctive, but that still obey the general rules of the interface. That is, things that look like folders open when I double-click them, and things that look like applications launch. In short, everything behaves exactly as it should.

. . . And Then There Were Themes

If you attended the recent Worldwide Developers Conference and caught a peek at Copland, you've noticed that future operating systems from Apple, in addition to providing a revamped standard look, will allow users to choose from several different interface "themes." One, for instance, is a "high tech" theme whose buttons have the matte black we associate with upscale stereo equipment. And a "Kids" theme is as friendly looking as KidPix. *[Editor's note: See page 8 of last month's issue for a look at the new Copland interface themes.]*

For years, users have been able to use various system extensions to customize the look of their programs, and certain developers have grown tired of waiting for Apple to set a 3D interface direction and have implemented their own interface looks. The result is just the sort of chaos you'd expect. Users could, for instance, have two or three applications open at once, and each one would sport a different look—along with different appearances for radio buttons, checkboxes, and other standard controls.

Themes give users a way to customize their systems, but it's a way in which the revamped standard look "makes sense"—each theme is internally consistent and obeys the overall rules of the interface. I'll admit that one of my biggest nightmares is that developers look at the new themes and say, "Hey! All the rules are gone! I can put any bloody kind of button in my interface I want to!" The new freedom that themes give users, however, actually puts more burden on developers to be consistent in their own implementations. Since users could change from the basic look to the high-tech look on a moment's notice, developers have to design their controls and screens in a way that will take on the appearance of whichever theme is active. For example, you can no longer assume the standard Macintosh look is running and paste a gray, 3D button in the middle of your dialog boxes—it will stick out like a sore thumb in some of the themes.

For as long as the Toolbox has existed, Apple has exerted its own "soft" constraints upon developers. It was much easier to do the standard thing than the nonstandard thing. For instance, you could put a standard scroll bar or button in your program with just a Toolbox call or two. On the other hand, if you wanted a mondo-cool 3D version of these, you were essentially on your own. This constraint acted as strong encouragement for developers to use standard controls. But the problems really began when either the need for new features or the desire to respond to customer demands for a more modern look

forced many developers to take the plunge and learn how to write the appropriate control and list definitions (CDEFs and LDEFs).

Luckily, while putting this new burden on developers, Apple finally developed an organized mechanism for extending the behaviors of buttons, lists, and so on, without forcing developers to toss them out entirely and write new LDEFs and CDEFs. Apple hopes that this will give developers a new way to make doing the right thing (sticking with the standard controls) the easy thing to do.

Till next time,
Doc
AppleLink: THE.DOKTOR

Got an interface gripe? Send it to Doc at AppleLink address THE.DOKTOR or Internet address the.doktor@applelink.apple.com.

OpenDoc

continued from page 15

would like to know whether this may present a problem for your part.

Q: The current selection model is annoying when you are doing things like page layout. How about if the resize handles were visible at the same time as the active frame border?

A: In today's applications, you must first select an object and then choose actions (such as Cut, Copy, and Resize) to perform on the object. That interaction standard is preserved in OpenDoc, and is appropriate because most tasks involve editing content, rather than resizing and moving entire parts. When you are manipulating entire parts (as in page layout), the page layout part may provide a "layout mode." As we've said before, modes should be accessed through tools or menu commands. Additionally, you should include some visual indication that the user is in a given mode.

Q: Why isn't there direct manipulation for scaling embedded parts? How about using "scaling" handles that are distinct from "cropping" handles? This might help users figure out what will happen if they move one of the handles.

A: The short answer is: Using different handles for different behaviors is a great idea. However, we recommend clearly delineating multiple functions. Therefore, we recommend not mixing different kinds of handles on a frame. Instead, let users access the different functions as separate modes, through commands or tools.

You may not be aware that both an embedded part and its container may cause the contents of a frame to be scaled (or rotated, skewed, and so on):

- If an embedded part's frame is resized, its editor must decide whether to crop or scale within the resulting frame shape. We strongly suggest that you use the cropping behavior as a default, rather than scaling. However, we realize that sometimes it may make sense to your part to scale content.
- A container must provide frame resizing (that is, cropping) for embedded parts. Additionally, a few containers (such as page layout or drawing) may also allow the user to change the rotation, scaling, and so on of embedded parts. To do this, we suggest your editor provide a tool or a command to enter a scale, rotate, or enter another mode. If you provide this capability through direct manipulation, you should change the appearance of the resize handles (to hollow squares or diamond shapes, for example) to provide visual feedback indicating that the user has entered a mode. At this point, when the user moves one of the handles, your editor would change the size of the embedded frame and change the external transform as appropriate (scale, rotate, skew, and so forth).

Q: During one of the demos at the OpenDoc Human Interface session (#213) at the WWDC, when a PICT part was dropped into a drawing part, not all of the contents of the dropped PICT were visible—they were cropped. Why?

A: When a part is added, it should ask its container for enough space to display its content. The container part dictates how much space the embedded part may occupy. If it isn't given a frame large enough to display all of its content, the embedded part should decide whether to display its content in a cropped format or, alternatively, request an additional frame (for example, a new page inserted in the document) from the container part. If the container part does not provide an additional frame, then the embedded part must crop itself. Generally the cropped content should be attached at the upper-left corner of the frame and cropped on the right and bottom edges as required.

Q: In the demo just mentioned, each time the three PICTs were resized, they appeared to "re-center" themselves. Is this a feature or a bug? I think it's strange behavior.

A: Oops! You caught us—the parts we used in our demo didn't display the correct behavior. The image should remain constant, and the user should be able to change the size and shape of the frame. The image shouldn't re-center itself when the frame is resized. Without this standard behavior, whenever a frame was irregularly shaped and then resized by means other than a corner resize handle, it would be difficult for the user to predict the result.

Q: It's obvious that when an embedded part is cropped (not all the contents of the part are showing), the content that is visible is what gets printed. But suppose the part has scroll bars, like a list of addresses. What should be printed?

A: In general, scroll bars are tools and, like other tools, should not appear in the printed document. However, this is up to you, the developer, to decide. We recommend that if the user can see the scroll bars on the screen, you print them. Otherwise, what is printed is different from what is on the screen. To allow the user to choose how to print, you should provide a command to show or hide the scroll bars, or provide a part editor preference for this purpose. A side note: If you follow the WYSIWYG model, only the contents currently visible (not necessarily all the contents of the part) should be printed. For more detail, consult the *OpenDoc Human Interface Guidelines.*

Q: How should my part print?
A: There are several aspects of printing. First, to basics: The Page Setup and Print dialog boxes contain options from the root part of the active window that apply to the entire document. We generally favor WYSIWYG, although there are exceptions depending on the kind of part. For example, movie and sound parts don't print very well.

If your part displays some static content, and it doesn't have scroll bars, you should print what the user sees on the screen. If the content is dynamic—for example, the part displays database query results— you should show the current value. Parts that have a preview or poster view, such as a movie, have the option of deciding to print that.

This brings up print options. Today's documents often have options for controlling printing. These may appear on a custom Print dialog box, or perhaps on a Page Setup dialog box, or sometimes on a Print Options dialog box. When your part is embedded, you should not display a dialog box at print time asking for options. You must follow the options of the document, or options previously set by the user. So how does the user set options on a part? This is done through Print Settings, which may be provided as a menu command. These should also be available in the Part Info dialog box by means of the Settings button. In addition, your editor may have printing preferences that apply to all parts that use your editor.

We will address more printing questions in future FAQs. ♣

QuickTime VR—A New Approach to Virtual Reality

By David Gleason

You can put away your 3D goggles and stop setting aside the cash you thought you'd need for developing virtual reality (VR) software products. QuickTime VR, released by Apple Computer, Inc., in May, is a new approach to virtual reality. Using QuickTime VR, you can create stunning, interactive virtual reality movies for business, educational, or entertainment CD-ROM titles; intriguing and effective in-house training materials; or customized visual guides for clients through three-dimensional spaces, such as office buildings or new aircraft.

With QuickTime VR, you can create CD-ROM titles that let your users explore faraway places, like the Mayan ruins of Chichen Itza in Mexico, or view the panoramas of Paris from the top of the Eiffel Tower and then zoom in on the Arc de Triomphe and Notre Dame Cathedral; you can let them wander and browse, away from the crowds, and without the cost and risk of travel. You can even give them the power to pick up and examine objects, from rare jewels and museum artifacts to the latest model (and still in production) Boeing 777 jumbo jet aircraft. You can create a title that follows the path described by Homer of legendary Greek hero Ulysses and the adventures that befell him on his way home to Ithaca, or show the site of the ancient city of Troy where the Trojan horse brought down the kingdom of Priam. And your users can view all these products on their Macintosh and Windows-based personal computers.

And, what is most significant to you as a developer, QuickTime VR offers faster development time, significantly lower cost, and a wider range of options than other virtual reality solutions. Unlike other VR solutions, such as Microsoft's highly-touted Surround Video, QuickTime VR does not require expensive and complex equipment such as a panoramic camera—you can take full advantage of the QuickTime VR technology using images from a standard 35-mm camera. As Rick LeFaivre, vice president of the Apple Advanced Technology Group, describes it, "QuickTime VR is virtual reality for the rest of us."

Built on the QuickTime 2.0 system software extension, QuickTime VR provides exciting new technology for the consumer and powerful authoring tools for the professional multimedia developer, on the desktop, without additional hardware devices or expensive, complex workstations. QuickTime VR provides a development environment that lets you create dynamic virtual reality products simultaneously for Macintosh and Microsoft Windows customers. Using QuickTime VR, you can author titles on the Macintosh computer platform and deliver this content to run on both Macintosh and Windows-based computers, exploiting the access that QuickTime VR provides to the vast market of personal computer users. LeFaivre says, "QuickTime VR could open up a whole new area of opportunity for Apple and for the developers who license the technology."

The rest of this article tells you how you can get QuickTime VR, what it is, and why it is a fast, inexpensive and versatile virtual reality solution. I then provide an introduction to the technology behind QuickTime VR and how you use it. Finally, I'll offer some insight into how some developers are using QuickTime VR already to rapidly create interesting virtual reality products, and offer some suggestions for how you might be able to use QuickTime VR to generate intriguing—and hopefully profitable—virtual reality titles yourself to an installed base of millions of customers.

What Comes With QuickTime VR

Beginning in May, Apple Computer began offering the QuickTime VR developer kit in two configurations that you can order from APDA:

- The QuickTime VR Authoring Tools Suite as a stand-alone product for $495 (U.S.).
- A bundle that includes the QuickTime VR Authoring Tools Suite, MPW Pro, and one seat at a QuickTime VR training course offered by Apple's Developer University. The bundle is priced at $1995 (U.S.).

Both products come in either PAL or NTSC formats.

Apple does charge a royalty fee for CD-ROM titles that use QuickTime VR: for shipments of CDs over 25,000 units, the charge is $400 for each set of 5,000 units sold over 25,000 units; for enhanced music CDs over 50,000 units, the royalty fee is $750 for each set of 25,000 units sold over 50,000 units. There are no royalties charged for noncommercial use of QuickTime VR.

The QuickTime VR Authoring Tools Suite (included in either configuration) provides authoring tools and run-time software that work in the MPW (Macintosh Programmer's Workshop) environment and permit you to create QuickTime VR titles for both Macintosh and Microsoft Windows platforms. The authoring tools suite version 1.0 comes with application programming interfaces (APIs) that permit you to create movies in MacroMind Director and stacks in HyperCard that can work with QuickTime VR movies.

The suite comes on four CDs—one with the tools and run-time software, and three others: a sample CD-ROM title that provides a tour of the Apple Company Store in Cupertino, California; a CD-ROM with all the development files for the sample title; and a CD-ROM with sample photographic images of the store. The Apple Company Store CD lets users visit the retail location and wander its halls, picking up objects and examining products on the shelves. Four volumes of documentation are included in the suite, plus technical notes, and a videotape, *Photographing QuickTime VR Scenes,* which provides visual instructions for planning scenes and shooting panoramas (the video is also available separately).

For more information on QuickTime VR licensing and developer support, see the news item on page 11 of this issue.

What Is QuickTime VR?

QuickTime VR is an addition to the QuickTime family, which runs on both Macintosh and Windows-based personal computers as well as upcoming consumer electronic devices such as game players and interactive television. QuickTime is a multimedia technology, supporting video, animation, sound, and text. QuickTime VR extends this technology, making it possible for you to create two types of movies that create the illusion of three-dimensional space—panoramic movies and object movies—which you create in the QuickTime VR interactive environment.

QuickTime VR provides what Apple is calling "image-based" virtual reality. This includes the following:

- camera rotation, by which the camera moves but the object it is filming remains stationary
- object rotation, by which the user is able to "pick up" and rotate an object using a mouse and keyboard
- camera movement, which means changing the user's viewpoint and viewing direction, permitting users to choose their own unique paths through the three-dimensional space
- camera zooming, which permits a gradual or rapid close-up approach to an object, or to draw closer toward or farther away from a scene, through image magnification and reduction

Together, these four elements of image-based virtual reality provide the basis for developing a wide range of user experiences. By combining interactive objects with panoramic scenes, and by allowing the user to look around and pick things up, QuickTime VR provides an experience that is very much like actually "being there."

QuickTime VR does not require a lot of time, or fancy graphics rendering. If you're on a limited budget but have the expertise and a vision of a great product, you can create exciting products in far less time and with less expense than on any other virtual reality platform.

Why Use QuickTime VR?

There are many interesting virtual reality products on the market today, but QuickTime VR provides advantages to developers that no other product can offer. It is a breakthrough product in its ease of use, lack of additional required hardware, and also in its 360° panoramic photography technique that lets you create visually impressive and custom-designed virtual reality products using inexpensive 35-mm photographs. Not only is the process potentially faster and less expensive than rendering images on a computer workstation, but 35-mm photographs can also provide greater detail and quality than rendered images. In addition, with a 35-mm camera you can use filters and different lenses to change lighting and create various effects.

QuickTime VR lets you create virtual reality environments by connecting a series of photographs in a realistic way; users can "walk" through the three-dimensional environment at their own pace, stopping to look at and even "pick up" objects, and clicking on designated "hot spots," which are interactive objects that you can define to permit the user to trigger some other action, such as a zoom in for closer examination.

Further, QuickTime VR provides many features not available on all virtual reality products, including full support for 8-, 16-, and 24-bit color, the ability to perform zooming and vertical panning, and variable viewing window sizes.

QuickTime VR runs on personal computers without additional hardware, and it opens the world of three-dimensional viewing to owners of low-end machines. All that is required is a medium-speed personal computer (for example, a Macintosh with a 68030 processor or better, or a 33-MHz 80386 PC running Windows 3.1), 8 MB of RAM (for a mid-range system), and a CD-ROM drive. Today, about half of all buyers of Macintosh computers include a CD-ROM drive in their configuration, and already there are about 2.5 million CD-ROM drives in use on Macintosh computers around the world. Add that to the many Windows-based personal computer owners who have CD-ROM drives, and you're talking about a market of many millions who can buy your virtual reality product and operate it without incurring additional costs.

All this is a powerful incentive for multimedia developers looking for a market for new products. Consider, for example, the highly successful CD-ROM title *Star Trek: The Next Generation Interactive Technical Manual,* which was the first title to use QuickTime VR technology, created by Imergy and published by Simon and Schuster Interactive. It uses QuickTime VR technology to let a user walk through the starship, and even pick up objects and

examine them closely. The CD was created entirely from 35-mm photographs (and a lot of production expertise). It took less than two months to shoot all the panoramic scenes on the Star Trek set. This shows that creating a successful, compelling product with QuickTime VR does not require a lot of time, or fancy graphics rendering. If you're on a limited budget but have the expertise and a vision of a great product, you can create exciting products in far less time and with less expense than on any other virtual reality platform.

Advantages of QuickTime VR for Developers

There are many virtual reality solutions in the marketplace today, and many different ways of presenting images in three dimensions. Compared to other virtual reality solutions, QuickTime VR provides several advantages for developers, including the following ones:

- *Easy panoramic image capture.* Panoramic cameras (which Microsoft Surround Video requires) are expensive, and panoramic images are difficult to digitize. With QuickTime VR, using a 35-mm camera to capture a full 360° panorama is an easy, low-cost alternative, yet 35-mm film provides high resolution and depth of image.
- *Faster performance.* QuickTime VR contains proprietary image warping technology that corrects for distortion of perspective while a camera is panning, and it is the first virtual reality product that is fast enough to make the photo-based experience realistic—and you currently can't get it anywhere but with QuickTime VR.
- *Small file size.* QuickTime VR compresses panoramas to as little as 540 KB each; this means, for example, that a CD-ROM with 600 MB storage capacity can hold more than 1,000 panoramas. By comparison, a single 360° view captured on video would require tens of megabytes.
- *Flexible interaction.* Hot spots are objects that a user can click on to move to another location or activate a different action; hot spots permit the developer to build interaction into any QuickTime VR scene, such as audio recordings, text, or linking to another panorama.
- *Extension of the QuickTime family.* Since it is built on existing QuickTime technology, QuickTime VR products can be made available to an audience of 10 million QuickTime-capable computers right now, on Macintosh and Windows platforms, and in the near future on consumer electronics devices such as game players and interactive TV.

Using QuickTime VR, you can create a wide range of products whose uses and design are limited only by your imagination.

Advantages of QuickTime VR for Users

QuickTime VR is not only an easier, more productive development environment; it also lets you provide a better virtual reality environment for your customers. There are three essential advantages that QuickTime VR provides for users:

- *Realistic simulations.* Virtual reality technology lets you show the world as it appears to the human eye, including correct perspective and rich detail, especially when 35-mm images are used. And the realism provided by the ability to pick up objects has until now only been available using expensive equipment.
- *Intuitive usage.* Unlike video and other three-dimensional display methods, QuickTime VR lets the user navigate through an environment and examine objects, as he or she chooses, using a simple and intuitive user interface. This includes the way that QuickTime VR indicates possible actions. (For example, the cursor changes shape when placed over a hot spot.)
- *Accessibility.* Requiring no special accessories, QuickTime VR can be used on low-end Macintosh computers and Windows-based computers with Intel 80386 processors or better. Any QuickTime 2.0–compliant application can work with QuickTime VR.

Advantages Over Surround Video

As mentioned in the February issue of *Apple Directions,* QuickTime 2.0 has a number of advantages over Microsoft Video for Windows; these advantages are worth repeating, including lower cost, more features, easier cross-platform development, and better performance. QuickTime VR maintains these same advantages over Microsoft's Surround Video, although they they may not be immediately apparent.

For starters, consider relative costs: Surround Video will be free to developers and will not require royalty payments. The QuickTime VR Authoring Tools Suite has a cost of $495, and there are royalty payments for volumes over 25,000 units for CD titles and 50,000 units for enhanced music CDs. On the other hand, Surround Video requires a panoramic camera for capturing three-dimensional images, and such a camera costs at least $2,000.

In addition, panoramic cameras only offer a single exposure setting within each panorama; if you are working with QuickTime VR and a 35-mm camera, you can get a better overall image by changing the exposure, as needed, for each 35mm frame. And unlike QuickTime VR, Surround Video provides no support for viewing interactive objects, no zooming or vertical panning, and it only supports 8-bit color. So for the cost of the suite of QuickTime VR tools and the royalty payments (if you produce enough volume to require payments), you get a much richer environment for creating more interesting products, and you don't need to buy expensive equipment.

Finally, Surround Video is slower and less effective on systems running Windows 3.1 than it is on Windows 95–based computers. In other words, Surround Video products will not have a substantial market until well after the release of Windows 95. QuickTime VR products will work—today—on any Mac OS–based or Windows 3.1-based system. (For more on the advantages of QuickTime VR over Surround Video, see "The Hidden Costs of Surround Video" on page 4 of the April 1995 issue of *Apple Directions.*)

In fact, with the speed of development that is possible using QuickTime VR, you could create a product starting *now,* and ship it and begin to generate revenue, perhaps even as soon as the end of this year.

Creating a QuickTime VR Title

Building a CD-ROM product with QuickTime VR requires a variety of skills and experience. A typical situation would include a team of multimedia professionals working together to produce a QuickTime VR title or project. Some of the required skills include the following:

- photography experience for capturing panoramas and images

Top: Three panoramic images of the Apple Company Store before stitching. Notice how the overlapped images do not properly match despite careful alignment. Bottom: The three images from the top illustration stitched together into one seamless image.

- Macintosh proficiency for working in the QuickTime VR authoring environment
- image-processing skills for performing post-production work on your images
- familiarity with MPW command-line syntax
- HyperCard and basic scripting skills for compositing QuickTime VR scenes
- multimedia project management experience
- working knowledge of Macromedia Director Lingo scripting language or HyperCard scripting

If you have a team in place that has these skills, you then can move into the authoring process, which begins with planning the project, determining the kind of experience you want to give your audience, deciding whether to use real-world photographic images or computer-generated images, whether to include objects that viewers can pick up and manipulate, and so forth. Your project may require storyboarding, floor plans and other standard video production techniques.

Creating Images With QuickTime VR

There are three basic procedures involved in creating a QuickTime VR movie:

- capturing the images with a 35-mm, video, or digital camera
- inputting images into a computer in digital form, generally using a scanner
- authoring/composing, which includes stitching together images, creating hot spots, and more

Most QuickTime VR productions employ these basic processes, although there are other options, such as including rendered images. And while it is possible to include panoramic images from panoramic cameras, it is not necessary; a 35-mm camera can provide all you need to create realistic and richly detailed three-dimensional images, because the QuickTime VR authoring environment can stitch them together for you.

Working With Panoramic Movies

To understand panoramic movies, it helps first to consider how a standard QuickTime movie is made since QuickTime VR is an addition to the QuickTime technology. A QuickTime movie is made up of one-dimensional compressed sequences, indexed by time. Each QuickTime movie may have multiple tracks, each track storing one type of media, such as audio, video, or text.

A QuickTime VR panoramic movie in QuickTime VR, by contrast, is multidimensional, event-driven, and spatially oriented. QuickTime 2.0 includes a new track type that was added for panoramic movies, where information about the panoramic movie is stored. The actual panoramic images themselves are stored in a regular QuickTime video track, which permits the developer to take advantage of the existing video processing capabilities and to build on experience with QuickTime to create QuickTime VR panoramic movies.

A panoramic movie is created in four steps:

- generating a panoramic image for each point in space from which the user will "stand" and be free to look around
- constructing "hot spot" images if desired
- linking the panoramas together (if there are more than one)
- dicing and compressing the images and hot spots to create the panoramic movie

The panoramic images can be created from still photographs, images from a panoramic camera, or rendered images; the total panorama can present a full 360° view from a given location or point. If still photographs are taken, the camera, mounted on a tripod, rotates in roughly equal (but not exact) increments.

Webster's dictionary defines *perspective* as "the technique or process of representing on a plane or curved surface the spatial relation of objects as they might appear to the eye." QuickTime VR lets you modify the perspective of a panoramic image through a process called *warping.* Warping is used to correct the distortions from the panoramic image, which in QuickTime VR is created in the form of a cylindrical projection (as opposed to a nearly flat screen for most computer monitors).

The images are then blended together in a process called *stitching,* which eliminates overlap and smoothes the borders between

individual images to create a continuous panorama. The QuickTime VR stitcher uses a proprietary algorithm to match and blend overlapping pictures; usually, about a 50 percent overlap is best for proper stitching. You can let the stitcher blend the images automatically, or you can use an interactive mode, which gives you greater control over the final stitched image. The result is a seamless, 360° panoramic PICT file, which can be further modified using any off-the-shelf paint program. See the figure on page 24 of the Apple Company Store for an example of individual photographic images that have been blended together to create a stitched image.

After stitching together the panoramic images, the developer can identify hot spots for particular locations. Hot spots are places where the user can interact with the scene, navigate to another location, or activate some action, such as causing an object to rotate.

The next step is to link the panoramas and hot spots together. If the QuickTime VR movie is to show a series of panoramas, additional images must be generated from each location.

Panoramic movies include *nodes,* each of which corresponds to a point in space where a 360° (or less) series of photographs was taken. A node also contains information about the node itself and links to other nodes. The nodes are stored in three tracks: one panoramic track that holds graphic information and pointers to other nodes, a first video track that holds the panoramic images for the nodes, and a second (optional) video track that holds hot spots.

Finally, panoramic movies and hot spot images are typically "diced" into smaller frames, called tiles, when stored on video tracks. This allows for more efficient use of memory, since only the diced tiles overlapping the current image are decompressed. See the figure "Creating a QuickTime VR panoramic movie" on this page for a diagram of the processes involved.

The outcome of the entire process is a set of linear QuickTime MooV files that can be played back tile by tile.

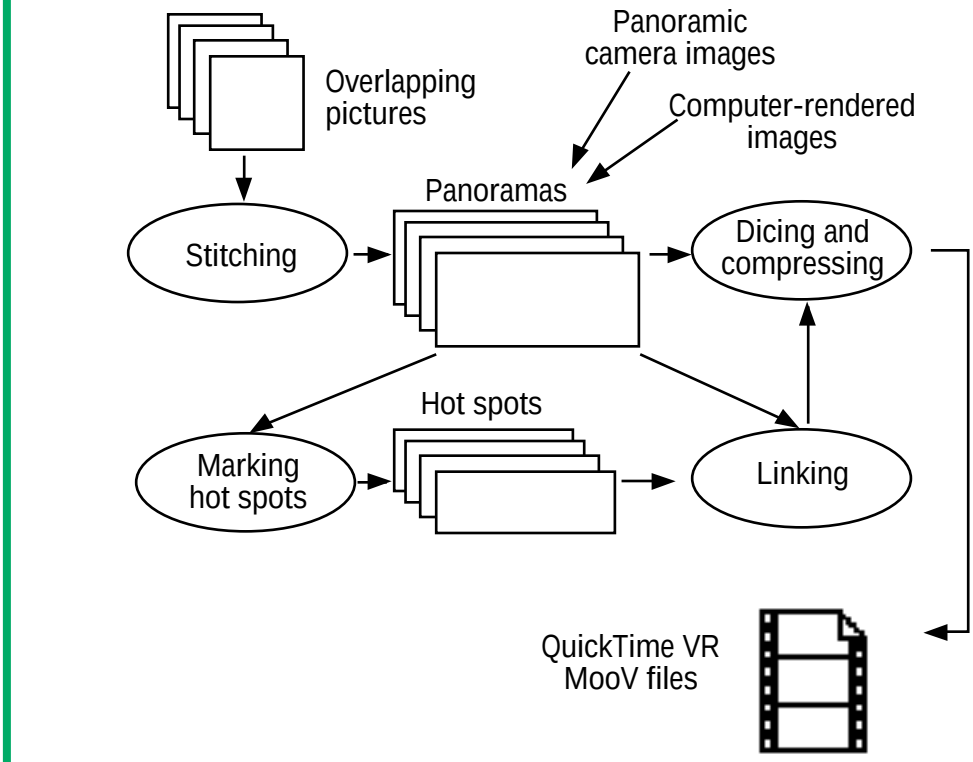
The QuickTime VR Authoring Tools Suite provides MPW scripts that let you create a single-node panoramic movie, with optional hot spots. You can also modify the user's default window size for viewing the panorama, or determine how far in or out the movie viewers can zoom.

The QuickTime VR Authoring Tools Suite also provides a scene editor that allows you to create multinode scenes. The scene editor is a HyperCard stack that lets you add, delete, and position nodes in a scene, create links between nodes for user navigation, and link interactive objects to nodes so they can be picked up and examined.

Working With Objects

An object movie lets the user examine an object from many angles, by picking it up and rotating it with the mouse or trackball. It gives the impression of rotating an object while holding it in your hand, turning it in different ways to examine all of its sides. If you have ever seen a priceless artifact in a museum, such as the Hope diamond or a pre-Columbian sculpture, you know the urge to pick it up and see it from many sides, which is what object rotation provides. A QuickTime VR movie can permit the user to rotate and examine the skeleton of a tyrannosaurus as easily as examining an artifact of fragile renaissance jewelry.

Typically, an object movie contains a two-dimensional array of frames, with each frame corresponding to a viewing direction. In filming, the camera is pointed at the center of the object and orbits around the object at constant increments. The camera can be connected to a frame digitizer inside the Macintosh computer, which synchronizes frame grabbing with the camera rotation. The camera can rotate as much as 140° vertically and the full 360° horizontally. If the camera moves at 10° increments, the result is



Creating a QuickTime VR panoramic movie. The photographer captures images with a 35-mm camera or other source; rendered images or images from a panoramic camera can also be included. The production team then digitizes the images, and stitches them together to create panoramas, which are then diced and compressed. Optionally, hot spots for user interaction can be added, and linked to the panoramas.

A 360° panoramic view of the Apple Company Store, made up of several images stitched together into one PICT file.

about 500 frames for a full range of rotation, up and down and across. The process normally takes about an hour to capture an object completely.

The object frames are stored in a regular video track, with other information stored in the movie header. For a real-world object that you have captured using a video camera, you can digitize and store images using software such as the QuickTime XCMD suite. For objects captured using a 35-mm camera, you can digitize the images with a scanner or have your negatives developed onto a Photo CD disk.

The next step may involve using the QuickTime VR scene editor to export the necessary data and tool calls from which you build your final QuickTime VR scene.

Developers Who Are Using QuickTime VR

What is it like to work with QuickTime VR? I talked with some developers who've had access to pre-release versions of the product, and their comments show the variety of ways they've come up with to use QuickTime VR to generate virtual reality products for their customers.

EF Education

A different application of QuickTime VR, by the EF Education Company, permits users to experience "language travel." The idea is to create an atmosphere, in a location in the country whose language is being studied, so that the student learns not only words and grammar, but cross-cultural experiences and communication as well. EF Education is using QuickTime VR to simulate a variety of interesting and attention-holding learning environments.

Bill Fisher, manager for EF Education, explains, "QuickTime VR has allowed us to quickly and easily capture real-world settings, complete with all their inherent cultural artifacts, and present to users a photo-realistic walk through an environment. Combining this with standard QuickTime movies allows us to simulate human-to-human interactions, and to draw users even further into the language learning experience."

An example of such a scenario, Fisher explains, is the location of diner, where a young woman sits in a booth, and some dangerous-looking men wearing sunglasses sit in a nearby booth. A waitress leans against a counter-top; there are other people in the background, as well as the sounds of a typical diner. The user then "sits" next to the young woman, and she says to the user, "Thank goodness you came. My name is Holly and I need your help." Holly begins her story, and the waitress interrupts to take the user's order. The user must make a menu selection to get the waitress to leave. The conversation continues, dependent upon the user's actions. Other scenes take place in an art gallery, at the beach, and so forth.

According to Fisher, "We are hoping to use QuickTime VR as a navigational tool, by modeling a three-dimensional topographical map as a VR object. I think this shows a lot of promise. And I think there are a lot of interface ideas beyond walkthroughs for which QuickTime VR could be really useful."

Nike, Inc.

There are many ways to use QuickTime VR productively; Nike, Inc., knows this and is using it in several ways—but so is the competition. As a result, Nike new technology consultant Dave Brunn is hesitant to discuss details of the products his company is creating with QuickTime VR. But he is willing to explain why QuickTime VR is an important element in Nike's commercial arsenal of technology. "We want to keep protecting 'the edge,'" says Brunn, "by not revealing exactly what we are doing with QuickTime VR or how we are doing it. But we want to convey to people that we are putting time and effort into working with this product."

Brunn says that the best part of QuickTime VR is how it "allows you to painlessly create virtual spaces to show concepts and ideas, in a relatively short time." Brunn is a photographer with a master's degree in fine arts, with expertise in digital and panoramic photography, so he understands the concepts and applications of this kind of technology. Brunn explains, "You can create three-dimensional space, depth of field, and perspective instead of just a flat image. In that sense, it is 'breakthrough' technology. You can load images into an application such as Adobe Photoshop, manipulate it, and it works well. I was a beta tester for Photoshop, and you definitely can do a better job of stitching together images with QuickTime VR."

The technology that went into QuickTime VR has made a positive impression on Brunn. He says, "The stitching is truly remarkable. You benefit from carefully positioning the camera, but it also makes up for a lot of problems in a remarkable way." Does QuickTime VR save time, cost, and effort? Doing it right, Brunn says, "takes a lot of planning, but it is easier and the results are very good. Planning a photo shoot helps tremendously; this permits you to do segments and navigation to a better degree. The software can do a lot, without a doubt; I've tried to push the limit, the bleeding edge, and I'm surprised at what it can accomplish. I've been experimenting with it to find the edge of stitching together photos, using blending, and so forth. Rendering the same images would take much longer, and would require more expensive and complex equipment and more hands-on work."

Perhaps best of all, Brunn says, "QuickTime VR works very well with the human mind, in a very natural way."

Possible Applications

Using QuickTime VR, you can create a wide range of products whose uses and design are limited only by your imagination. The

following areas offer excellent opportunities for you to create commercially viable products for consumers or for specific clients.

• Education offers an almost limitless realm of opportunity for QuickTime VR products, from CDs on science and nature to art and history, interactive educational games, and advanced interactive courses for higher education. QuickTime VR's ability to display rich visual images is a major benefit that until now could only be produced with expensive equipment.

• Engineering provides opportunities to create titles that let the user analyze and explore building or product design or teach manufacturing or construction principles interactively—for example, designing and building a new automobile with a variety of options, such as different body shapes, colors, and accessories. Photographs of actual parts or other objects can be made into PICT images, and then modified or colorized in a standard paint application to show a wide variety of options, without resorting to time-consuming computer rendering.

• Museums will probably be a major beneficiary of developer attention, because of the ability of QuickTime VR to permit the user of a museum CD title to walk through an uncrowded tour of the Prado in Madrid, or the American Museum of Natural History in New York, looking at paintings, ancient masks and weapons, sculpture and tools, picking up objects and examining them more closely than is possible in the museum itself.

• Kiosks can provide a quick orientation to a shopping mall or theme park using QuickTime VR technology. Shoppers can click on store icons and examine merchandise or simply "look around" the clothing department to see what brands are carried and how clothes are displayed.

• Architects can benefit from providing customers with tours of buildings under construction, a model home, or a QuickTime VR custom title filled with images of design elements, such as fabrics, tiles, wood paneling, kitchen and bathroom fixtures—everything that goes into creating a home or office building.

• Travel can be simulated using QuickTime VR movies that tie together 35-mm photographic images of exciting locations and adventurous journeys, from hiking in the Himalayas to participating in a photographic safari in East Africa. Travel agencies would use such titles to show potential clients the highlights of a proposed tour, where they would stay, even some sample meals and a tour of the local market.

• Games are a vast realm ready for exploration, since realistic, detailed photographic images can lend realism and excitement to a game's landscape or interior setting. QuickTime VR technology lets the user explore and wander at will, and interact with objects, for example, picking up and examining clues in a criminal mystery title.

These are simply some suggestions for how you could use QuickTime VR to create your own products; you and your multimedia team will probably come up with ideas no one else has thought of yet, and in that may lie your greatest opportunity. ♣

David Gleason is the president of Verona Communications, a World Wide Web service provider, based in Palo Alto, California. He is also a frequent contributor to Apple Directions *and a former localization manager for Apple Computer Europe.*

Ask Don Norman

Where Will Apple Be in Five Years?

This year's Worldwide Developers Conference ended on an intriguing note, with Apple Fellows Don Norman and Alan Kay participating in a round-table discussion with Apple CEO Michael Spindler on the subject of the future of computing, moderated by Apple Advanced Technology Group Vice President Rick LeFaivre.

Alan Kay raised a very interesting possibility: That our culture is in a transition that will soon result in widespread computer literacy, in the way that changes in education and the introduction of the printing press led to the widespread ability to read and write in the 16th and 17th centuries. What Alan means by computer literacy *is not just the ability to use computers, but the ability to program them, as well, at least to some degree.*

Before most people could read and write, he says, most writing was done by professionals; after literacy became commonplace, though, most writing was done by nonprofessionals. Similarly, most current programming is done by professionals (like yourself); once computer literacy becomes the norm, Alan thinks that most programming will be done by nonprofessionals as they figure out how to use computing technology to best augment their daily lives.

The session ended with Rick LeFaivre asking the question "Where will Apple be in five years?" Don Norman returns to the pages of Apple Directions *this month to sum up the panel's answer to that question. Don invites your questions, which he's agreed to answer in print on an occasional basis; you can reach him on AppleLink at DNORMAN.*

Where will Apple be in five years? Nobody can predict the future, but let me make some intelligent predictions. Caveat: What follows are my own, unofficial speculations: I could be wrong. My speculations are not to be taken as an official statement of Apple Computer, Inc. But, obviously, these predictions are consistent with our existing strategy.

First, let me outline what I think are four generations of the personal computer.

First Generation: Apple II and IBM PC

Small, relatively inexpensive. Not very powerful, not very easy to use. Nonetheless, the first step toward making computers accessible to everyone. Their real significance was that they existed; they proved that they could be built and that they could perform useful services.

Second Generation: Graphical User Interface

Here the emphasis was on ease of use. The graphical user interface, or GUI, was first introduced commercially outside of laboratory

and research computers by Apple, with its Lisa and Macintosh systems, following the innovative work at Xerox PARC and university LISP machines. Today, GUIs are commonplace, and Windows 95 will complete the transition for all but UNIX diehards. With Windows 95, Microsoft joins the second generation of personal computing just as Apple moves ahead to establish the third-generation platform.

Third Generation: Component Software on RISC Hardware

Apple is starting this new generation with the introduction of Power Macintosh RISC hardware and OpenDoc software. This is the generation that develops software that fits the way people work. Component software allows developers to produce custom solutions for their customers, who can then put together containers and components in whatever way best fits their needs and preferences. The look and feel of the computer can be varied. No longer will the OS and applications have a "one-size-fits-all" look. Instead, there will be a wide variety of software solutions to fit every requirement, every market segment.

The "ease of use" slogan of the second generation actually illustrates the problem: Complex second-generation systems had to be made "easy" before everyday people could use them. The third-generation systems fit the tasks and lifestyles of the users. Ease of use will no longer have to be stressed, for it will be considered a given.

Fourth Generation: Component Hardware

Just as the third-generation system is component software, the fourth generation will be component hardware. What do I mean by this? We have to keep some secrets, but look at Newton and Pippin for some hints.

Where Will Apple Be?

Returning to the question, in five years Apple will have moved to the next step in the transition from the second generation to the third. The transition requires two steps, the first being the hardware transition to RISC machines, the second being the software transition to components (OpenDoc).

The hardware transition is underway. Soon, all of Apple's platforms will use RISC CPUs, from the least expensive home devices to the most powerful desktop mammoths and servers. Power Macintosh systems and Newton devices are already driven by RISC technology. Forthcoming set-top boxes, special-purpose entertainment and home devices, and new devices for home, business, and education will also use RISC processing.

The second step in the transition to the third generation is the widespread adoption of component software, by which I mean the transition to OpenDoc. Within five years, OpenDoc will pervade all aspects of Apple's business. The OS will be built around components. Most applications will be done in a pluggable, component manner; the days of the old, monstrous, monolithic application will be over, although legacy applications will still be used by an ever-dwindling set of people. (Remember, some people are still using five-year-old applications today. In fact, there are many perfectly happy Apple II users!)

All of this will have a profound and positive impact upon ease of use, flexibility of performance, and the ability of users and service providers to develop their own custom solutions—millions of solutions. This will also have dramatic and positive impact upon the business side of software. Component software will empower independent developers who, today, cannot compete with the big guys.

Within five years, OpenDoc will pervade all aspects of Apple's business. The OS will be built around components. Most applications will be done in a pluggable, component manner; the days of the old, monstrous, monolithic application will be over… .

Programming by Everyone?

To take up Alan Kay's answer to the same question—that in five years the world will be computer literate and more programming will be done by nonprofessionals—I'd like to comment on what that means for you developers.

In short, that should be mainly good news for you.

First of all, not everyone will be computer- and programming-literate. The nonliterate 40-year-old of today will be a nonliterate 45-year-old. Even in the most optimistic of cases, that 40-year-old illiterate will be semiliterate in five years.

What about increased spread of programming ability? Well, I'm suspicious. As Alan has pointed out in other contexts, we still don't have much of a spread of mathematical or scientific literacy, so why should we expect anything different with programming? Besides, although I'm a fan of increased ability to process information and understand computation, I don't see much virtue in widespread teaching of today's limited, ill-constructed programming languages. Finally, even if lots of 10- to 20-year-old-kids today are learning programming, it simply means that in five years we will have more 15- to 25-year-olds who can program. That's still a minority of the population.

Ability Doesn't Equate With Skill

But let's say that lots more people gain the ability to program in the next five years. I'll use myself to illustrate what I think will happen. I can program (badly) in more languages than make sense. Here's a partial list of the languages, in the order in which I learned them: Remington Rand UNIVAC machine language (*not* assembly, but real machine code); DEC assembly language for the PDP-1, PDP-4, and PDP-7; FORTRAN; BASIC; SNOBOL; LISP; C; Prolog; UNIX C-shell; HyperTalk; and AppleScript.

All those languages, yet today I am fluent in none of them. And the fact that I do not write code today is my preferred state, thank you very much. I prefer that professionals do the development. My knowledge of programming makes me a better user, makes me better able to appreciate and understand what I am using and doing, but the difficult job of creating a workable set of applications—or OpenDoc parts and containers—will always be difficult.

Even when we have the world's most perfect development tools, it will take the developer's skills and talents to do great products. Writing good programs is difficult. It is a skill

that takes years to develop. Knowing *how* to program is only the first step.

That being said, there still is an important role for user programming. No application, no matter how cleverly designed, can fit the myriad demands of the user population. Systems that allow users to tweak them, to construct simple scripts and structures to make the developer-provided tools better fit their needs, will allow users to make dramatic increases in power. Thus, developers can provide tools to allow users to do some final programming—esepcially scripting—of the end products.

Compare programming with writing. Almost everyone is literate and can write, but I prefer to read books written by professionals, not by the average person. The fact that everyone can write is of great practical value in life, but in no way has it reduced the role of the professional writer. The skill of storytelling, or of clear, concise writing, requires much more than the ability to put words on paper. The same holds for programming. Computer users will require the expertise of you professionals for a long time to come. ♣

CD Highlights

continued from page 14

you to create, display, and edit documents in Japanese on the Macintosh computer and other computers running the Mac OS.

Enhancements in this version of the Japanese Language Kit include System 7.5 and QuickDraw GX compatibility, a TrueType version of the Osaka font, an updated input method, and system resources that let WorldScript run in "native" PowerPC mode on Power Macintosh computers.

PowerPC Hardware Reference Platform

This document describes the PowerPC Hardware Reference Platform (HRP) proposed by Apple, IBM, and Motorola. HRP is the proposed hardware standard that will allow anyone to create PowerPC processor–based computers capable of running the Mac OS and several other operating systems.

Toolbox Assistant Update #1

This is the first update to the Macintosh Programmer's Toolbox Assistant. It contains two new databases and one updated database along with a new version of the QuickView application. See the associated Read Me files for installation information.

Tracer

PowerTracer is a system ATrap and PowerPC function tracing and timing tool. Its objective is to show calling patterns and timing information for the Macintosh Toolbox. PowerTracer patches a very large list of system ATraps as well as functions in the PowerPC public interface library. The tool configures itself appropriately on 680x0 Macintosh computers as well as on Power Macintosh computers.

PowerTracer gives detailed trace output information, including the execution architecture for each system function (that is, 680x0 or PowerPC), execution architecture for the caller of a system function, and interrupt information. Trace information is collected in memory and spooled to disk asynchronously.

Coming Next Month

Yes, I know I promised you new printer drivers on this disc in last month's column; I'm pretty sure I'll be able to squeeze them onto next month's disc, along with the usual assortment of new tools 'n' stuff.

Alex Dosher
Developer CD Leader

Business & Marketing

Market Research Monthly

Macintosh User Profile Research From the WWDC

More Than 4.5 Million System 7.5 Users

Normally this column takes some small amount of data and tries to point you to a market opportunity, or highlight Apple platform successes in the marketplace. In a slight departure, this month we bombard you with data from Apple's most recent proprietary Macintosh User Profile study, completed in 1994. We're doing this because the data was handed out to attendees at the May Worldwide Developers Conference (WWDC), and we wanted to make it available to the entire Apple developer community, whether or not you attended the conference.

Apple also released data at the WWDC that was not included in the User Profile Study. Most notably, as of the mid-May WWDC, more than 4.5 million Macintosh customers are already using Macintosh System 7.5, the latest version of the Mac OS, which was released just last September. That's nearly 25 percent of the current worldwide Macintosh customer base. By comparison, the last major system upgrade—Macintosh System 7.0—was adopted by just over 15 percent of the customer base in its first eight months on the market.

Accounting for the rapid upgrade to System 7.5 is the current large volume of Macintosh—and especially Power Macintosh—shipments; all Macintosh systems have shipped with System 7.5 since last September. In both fiscal quarters since the System 7.5 release, Apple has shipped record numbers of Macintosh systems, more than 2 million in each quarter.

What this means for you is that an increasingly large share of the Macintosh installed base now has access to the advanced features

Inside This Section

Apple's 1994 User Profile Study

	Small business	Medium/large business	Home office	Home	K–12	Higher education
Which equipment do you use?						
Modem	47%	34%	53%	45%	24%	31%
CD-ROM	19%	20%	20%	17%	33%	15%
Scanner	27%	11%	18%	6%	19%	8%
Do you use a color printer?						
Yes	12%`	19%	14%	16%	22%	3%
No	78%	73%`	83%	78%	67%	87%
Don't use printer	10%	8%	2%	6%	12%	9%
Do you fax documents from this Macintosh?						
Yes	26%	17%	30%	17%	10%	14%
No	74%	83%	70%	83%	90%	86%
RAM size						
1 MB	–	–	–	3%	–	5%
2 MB	6%	–	4%	9%	16%	14%
3 MB	–	–	–	3%	6%	–
4 MB	13%	11%	12%	37%	15%	35%
5–6 MB	6%	3%	13%	11%	24%	3%
8 MB	44%`	51%	39%	20%	21%	22%
9–10 MB	4%	6%	11%	8%	9%	8%
12 MB	9%	3%`	5%	4%`	6%	3%
Is computer 32-bit capable?						
Yes	68%	83%	92%	79%	84%	79%
No	32%	17%	8%	21%	16%	21%
Does computer have stereo capability?						
Yes	70%	76%	65%	47%	53%	61%
No	30%	24%	35%	53%	47%	39%
Is stereo sound combined for internal speaker?						
Yes	28%	48%	45%	23%	19%	44%
No	72%	52%	55%	77%	81%	56%

Is sound input device present?						
Yes	50%	68%	78%	66%	63%	66%
No	50%	32%	22%	34%	37%	34%
Can computer play and record together?						
Yes	24%	35%	35%	18%	12%	34%
No	76%	65%	65%	82%	88%	66%
Can computer play and record 16-bit audio?						
Yes	2%	4%	5%	3%	2%	1%
No	98%	96%	95%	97%	98%	99%
Does Sound Manager support multiple channels?						
Yes	10%	10%	10%	13%	5%	12%
N/A	90%	89%	90%	87%	95%	88%
Does Sound Manager support 16-bit audio?						
Yes	10%	10%	10%	13%	5%	12%
N/A	90%	90%	90%	87%	95%	88%
All monitors—screen size						
1024 x 768 (19")	8%	6%	9%	–	–	–
1152 x 870 (20/21")	4%	4%	2%	1%	–	1%
512 x 342 (built-in)	16%	6%	6%	21%	21%	22%
512 x 384 (12")	10%	3%	14%	12%	26%	14%
640 x 400 (PowerBook)	3%	15%	8%	5%	4%	17%
640 x 480 (13/14")	46%	51%	50%	56%	48%	33%
832 x 624 (16")	5%	14%	5%	4%	–	9%
640 x 870 (15")	12%	7%	10%	2%	3%	5%
All monitors—maximum pixel depth						
1 bit	16%	11%	6%	5%	11%	15%
2 bit	–	1%	–	–	–	3%
4 bit	4%	9%	10%	6%	4%	13%
8 bit	53%	59%	51%	49%	53%	48%
16 bit	11%	9%	20%	19%	20%	9%
32 bit	9%	9%	10%	2%	2%	–

offered by System 7.5, including Apple Guide, QuickDraw GX printing and graphics, and PowerTalk. You can differentiate your product, and make it appealing to this large group of customers by adopting System 7.5 technology. For details about System 7.5, see "System 7.5: Apple's Unified Operating System for 680x0 and Power Macintosh Computers" in the June 1994 issue of *Apple Directions*.

Additionally, Apple disclosed the following at the WWDC:

- 11 percent of Power Macintosh buyers are replacing a DOS/Windows PC.
- 29 percent of Macintosh systems acquired in October through December of 1994 were purchased by first-time Macintosh buyers.
- 54 percent of all Macintosh customers use two or more Macintosh systems.

The accompanying chart, "Apple's 1994 User Profile Study," provides the rest of the data handed out at the WWDC. We'll leave interpretation of the data to you this month. We hope that it will help your organization plan product releases that are well tailored to the customers you're trying to reach. ♣

Marketing Feature

The Macintosh Entertainment Market Takes Off

By Eric Klein, Entertainment Evangelist, Apple Computer

As the entertainment evangelist at Apple Computer, Inc., I'm lucky enough to be able to work with some of the most creative people in the software business—game developers. I provide these developers with technical and marketing assistance in developing games and entertainment titles. And this experience has taught me a lot about what it takes to create a product that will fly in the Macintosh entertainment market.

The entertainment market is growing at an incredible rate right now, but not just any game sells well to Macintosh users—the formula for success is complex and ever changing. If you're new to the entertainment business or the Macintosh platform, it's wise to learn from those who have come before you. Herein lies this article's purpose: It provides you with a snapshot of the Macintosh game market, insights on the expectations of Macintosh game players, and information on Apple products, technologies, and marketing programs that will influence this business—all of which should help you create better, more successful Macintosh entertainment products.

A Snapshot of the Macintosh Game Market

The Macintosh game market took off like a rocket last year. Industry-wide sales of entertainment software went up 55.9 percent in 1994, surpassing all other categories of software. And Macintosh entertainment title sales far exceeded this industry average, growing at a rate of 83.0 percent. (See the chart on page 30 for a detailed breakdown of these Software Publishers Association [SPA] figures.)

With this growth rate, Macintosh developers were pleasantly surprised by last year's sales figures. And developers in the saturated DOS-compatible market are seriously looking at creating Macintosh versions as a way to squeeze more revenues out of existing titles.

Investors and venture capitalists have also taken notice of

these growth figures, and many are proactively looking for game companies to invest in. At April's Spring Computer Game Developers' Conference, a venture capitalist's view of the market was provided by Mark Gorenberg, a partner with the Emeryville, California, firm Hummer, Winblad Venture Partners: "This is a great time for new software game companies because about a third of U.S. homes now own a computer, CD-ROM adoption is still growing well beyond estimates, and there are no dominant players in the games category to impede newcomers. We encourage all of our companies to develop for multiple platforms, including the Macintosh, which has been traditionally strong in the home-based entertainment and education markets."

Another attractive aspect of the entertainment software market is that it has the capacity to expand beyond the installed base of computers. Games are a lot like chocolates—once computer owners get the taste of one, they just want more. This trend was recently backed up by an SPA consumer survey of home computer owners, which found that a third of the households that used computer games had more than *ten* games. (See the chart on page 33.)

The Market Challenges

Of course, I wouldn't be doing my job if I didn't talk about the risks of entering this market. Hummer, Winblad's Mark Gorenberg sums up these risks: "Game developers are often at the mercy of bleeding-edge technologies and limited retail shelf space, and since this is a hits-based business, you're only as successful as your next game."

But in this strange business, where you may work 12 to 18 months on a game with a shelf life of only 12 to 18 months, the philosophy of "the journey is the reward" definitely applies. The pursuit of a hit game—that elusive combination of plot, game play, graphics, and music—is probably what keeps many successful game developers coming to work every day.

Doug Grounds of Lion Entertainment, developer of the Macintosh versions of Super Wing Commander and the upcoming Doom II (GT Interactive is the publisher), talks about what makes a hit game. "You have to approach your game project like a Hollywood blockbuster movie—the entertainment experience has to draw your users into another world. You know you've done your job if users forget about the computer and your interface and are just left with the reality and challenges that you've created for them."

One way to minimize risk in this business is to deploy your games across more than one platform. And I'm sure you won't be surprised by my strongly held belief that developing for the Macintosh platform provides a great entertainment business proposition. Why? First, the Macintosh has a rapidly expanding installed base of users who tend to buy multiple games. Second, Apple provides great cross-platform development technologies that enable developers to author visually and technically appealing games that are easy to move over to other computer systems. And third, because of the Macintosh platform's consistency and ease of use, it costs you less to support Macintosh-compatible products.

Apple's Consumer Focus

Santa was good to Apple's customers last holiday season, delivering more Macintosh computers (primarily via the Performa line) to home users than ever before. These home user sales turned out to be a boon to Macintosh entertainment developers as well, given that more than three-quarters of these users are likely to buy entertainment software (source: March 1995 SPA Consumer Survey).

Virgin Interactive Entertainment, a $200 million interactive media company, recently began marketing Macintosh products because of the attractive demographics of the platform. Their first Macintosh release, The 7th Guest, has been a runaway hit. Rand Bleimeister, their senior vice president of sales and marketing, comments, "The best-selling Performa line helps us reach an important secondary game market—boys under 18 years old."

Apple's own research on Performa computer owners provides more insight into why game developers love this product line.

- 76 percent of users have an income of more than $35,000 per year.
- 55 percent of all Performa users have children under 18 years of age.
- 77 percent of Performa users say that they use their computers for entertainment, and 78 percent use them for their children's education.

One of the reasons that Macintosh computers have been selling so well into this market is that these systems are multimedia-ready and easy to use. A 1993 study by Dataquest named Apple Computer, with 2.4 million multimedia-ready computers installed, the leading worldwide multimedia personal computer vendor. (Dataquest defines a multimedia personal computer as one that has a CD-ROM player and sound capabilities.) And according to SIMBA Information, Inc. (1994), this gives Apple 33 percent of the existing installed base of multimedia-ready personal computers.

The Power Mac Impact

I could, of course, fill the next three pages with market research data, but why don't I skip to the heart of the matter: What really gives game developers and players that gleam in the eye is the full-throttle speed of the Power Macintosh. Not to mention the Mac OS technologies that enable you to go where no other game developer has gone before.

Bungie Software is an eight-person Macintosh-only game developer best known for Marathon and Pathways into Darkness. Alexander Seropian, CEO and cofounder, started this company four years ago while he was a University of Chicago student. He credits his company's success to the technical wizardry of its games, its mastery of low-cost guerrilla marketing tactics, and a growing demand for good Macintosh games.

"Our game sales have grown exponentially in the last four years. By the time Marathon shipped this December, we had presold more copies than the total sales of our two previous

Entertainment Software Sales Growth

Comparison of North American Microcomputer Software Sales

	(Dollar sales in millions) Overall	Macintosh
1993 sales	$459.3	$51.1
1994 sales	$716.2	$93.6
1993–1994 change	55.9%	83.0%

Source: SPA, April 1995

games combined," says Seropian. "I think the Power Macintosh is the most significant factor behind this demand. The technology has made it possible for home users to have cool games, and growth in the installed base has helped give the Macintosh platform credibility among developers."

Let me expand on the technical aspects of the Macintosh that make it a superior platform for game development.

• *Multimedia readiness.* When I help DOS game developers move their products over to the Macintosh, they're almost always impressed by the built-in capabilities of the Macintosh. With every Power Macintosh, users get a minimum of 8 MB of RAM, built-in networking, QuickTime video support, and 16-bit stereo and MIDI sound support. In addition, almost 90 percent of Power Macintosh models ship with double-speed CD-ROM drives today, and this summer, models will ship with quadruple-speed CD-ROM drives. All this adds up to significantly lower compatibility testing and user support costs.

• *Awesome speed.* In your pursuit of killer games, speed is of the essence, and the Power Macintosh delivers it. Benchmark testing from independent sources confirm that Power Macintosh computers are much faster than their Intel-based counterparts. What's more, the PowerPC floating-point capabilities are nothing short of awesome. Another transition that should help your need for speed is the conversion from the NuBus expansion slot standard to the faster, more mainstream Peripheral Component Interconnect (PCI) standard. The PCI standard has the potential to provide users with much higher performance in accessing graphics acceleration, video, and networking cards.

• *Rich cross-platform development technologies.* QuickTime, the QuickTime Music Toolkit (available this summer), QuickTime VR for 3D environments, and upcoming technologies, such as QuickDraw 3D and Open Transport, help you leverage your game development costs by making it easier to deploy your games across multiple platforms. The next technology to be released, QuickDraw 3D, is a three-dimensional extension to the QuickDraw graphics architecture that will provide Macintosh-based and Windows-based computers with integrated system support for 3D objects. This Toolbox extension includes an optimized renderer, low-level access to 3D accelerators, and a 3D graphic primitives library.

Invasion of the Cross-Platformers

These advantages—installed base growth and technologies that have made Macintosh entertainment developers see dollar signs—have also attracted the attention of game developers on other platforms. Of late, the roster of game developers and publishers new to the Macintosh platform include the likes of LucasArts (X Wing, Dark Forces, Rebel Assault); Access (Links Pro); Activision (Return to Zork); Virgin Interactive (The 7th Guest, The Daedalus Encounter); GT Interactive (Doom II); Digital Pictures (Slam City); Origin (Super Wing Commander); Novalogic (Wolfpack); and many more. Also, major joystick vendors (Thrustmaster, Advanced Gravis, MicroQue, and CH Products) have begun building input

Entertainment Market Resources

The following resources are currently available to help you learn more about developing and marketing entertainment software.

• "The Games Marketing Game—Playing to Win in the Entertainment Software Market," *Apple Directions,* July 1993, page 23.

• "Ten Tips for Games Developers," *develop* Issue 17, March 1994, page 114.

• "Pippin: A New Platform for Multimedia Titles," *Apple Directions,* May 1995, page 17.

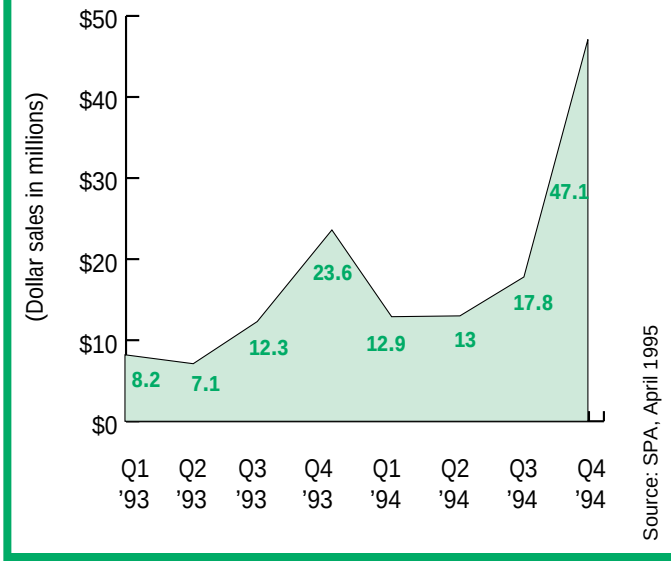
• *Tricks of the Mac Game Programming Gurus* by Bill Dugan, et. al., Macmillan Publishing (ISBN 1568301839), $50.00. This book covers all aspects of game programming and includes a CD-ROM packed with various tools, libraries, utilities, sample code, and games. Contributors to this book include some of the top game programmers. Though this book won't be available until August, you can order it now by calling 800-662-3058 or 317-581-3500.

• *Developer Game Kitchens.* Several times a year, Apple sponsors local game "kitchens," programming clinics where Apple engineers help you port or perfect your Macintosh game. As you might expect, the competition for kitchen slots is fierce, but if you'd like to be considered, send a persuasive e-mail to Eric Klein at klein@apple.com.

• *The Computer Game Developers Conference.* This annual interactive entertainment conference of lectures and round-table discussions aims to foster information exchange among computer and video game industry professionals and to enhance the quality of entertainment software. To get a copy of their informative April 1995 conference proceedings, or to receive information for next year's conference, which will be held in late March of 1996, call 415-948-2432.

• The Red Herring *venture capitalist list.* Every December this venture capital magazine publishes a list of "VCs" active in education and entertainment technology. This magazine also sponsors a business plan fair in January. For a fax of the VC list, call 415-865-2277.

Macintosh Entertainment Software Growth: North American Sales, 1993–1994



devices for the Macintosh.

Rand Bleimeister of Virgin Interactive recently talked about the business case for moving to the Macintosh platform: "Entering the Macintosh market was a low-risk business decision for us. Today it costs us approximately $1.2 million to bring a Windows-based game to market. By spending an additional $100,000 to $200,000 to move a game to Macintosh, we know we can get a great return on our investment. What's more, we know we'll spend less supporting customers over the life of the product."

GT Interactive, publishers of the monster hit Doom II, will soon release a Macintosh version. "We sold one million copies of Doom II in the first six months," says Allan Blum, vice president of marketing, "and we had so many requests for it on the Macintosh platform, we felt we just had to move it over."

One thing that game developers new to the platform are learning is that Macintosh users have higher standards for usability, configurability, and ease of instal-

The Seven Deadly Sins of Macintosh Game Development

My observations on what makes a best-selling Macintosh game are a bit simplistic: A great game needs to be unique in some way, addictive to play, and high in quality. And while this advice falls in the category of "easier said than done," I can also offer you more tangible advice on what I know *doesn't work* in game development. In my experience as Apple's entertainment software evangelist, there are seven "deadly sins" that can, at a minimum, diminish product sales—and, at worst, can cause the early demise of your product. Here are my seven deadly sins of game development.

1. Developing for the Macintosh as an afterthought. For those developers entering the game market for the first time, I wholeheartedly recommend a common code strategy for cross-platform development—in other words, keeping a single code base under development for as long as possible before branching off and customizing the code for different platforms.

Developing for the Macintosh as an afterthought almost always results in an inferior Macintosh product that doesn't sell well. Because the Macintosh has a richer set of multimedia technologies, it's better to aim high at the start of a project, by creating rich features and media components (such as separate music and voice tracks and high-resolution animations), than to simplify the Macintosh version for other platforms. (For advice on finding expert animators for your games, see the next article, "Recruiting Traditional Animators for Multimedia Projects," on page 34.)

On the benefits of cross-platform development, Loren Cobb, president of Corrales Software Development (a developer of war games and simulations for national and international clients) adds, "The exercise of writing for multiple platforms costs almost nothing compared to the benefits. And when you develop for multiple platforms, you end up with a better product—the bugs and weaknesses that are revealed when code is compiled on different compilers is astounding."

2. Writing in assembly code to boost performance. Though you may be tempted to increase your product's performance by writing in assembly code, be warned that this will probably make it much harder for you to move your products to other platforms. Instead, I recommend that you write in C or C++, writing in assembly code only in key areas where speed is crucial. And by all means tap into the incredible speed of the Power Macintosh computer's floating-point processor, and use this computer's many general-purpose registers for more efficient memory management.

3. Bypassing Toolbox routines. Game developers are notorious for working around the Toolbox APIs (application programming interfaces) of the Mac OS and writing directly to the hardware in order to boost performance. Avoid this approach at all costs, because writing to the designated interfaces will keep your product from "breaking" as Apple rapidly evolves its hardware. If you must write around Toolbox routines for speed, create a redundant routine that accomplishes the same task using the Toolbox. Then add a User Preferences checkbox that enables users to choose between faster or more compatible routines.

4. Porting to the Macintosh without modifications. A lesson that many DOS, Windows, and game-player developers learn the hard way is that Macintosh users are more demanding than other users. Macintosh users have come to expect high-resolution graphics and great sound. And if they've gone to the trouble of purchasing a Power Macintosh, they want to "feel the speed."

My advice to multiplatform developers is to take the time to create high-quality 3D-interface Macintosh graphics, then move to them to lower-resolution platforms through a good cross-platform standard like SVGA. Optimize Macintosh performance if you're moving code over from a DOS or Windows game, and write a Power Macintosh native version of your product soon.

5. Ignoring Macintosh interface guidelines. Moving a DOS interface, unchanged, over to a Macintosh game is the kiss of death for that product.

"Make your application Mac–OS friendly or you'll get nailed by reviewers and users," says Jake Hoelter, executive producer at Presage Software Development Company (Macintosh versions of Prince of Persia and Lemmings). "We try to follow the spirit of the guidelines, if not the letter."

Customers have come to expect that all their Macintosh products will have the same "look and feel." Buy the book *Macintosh Human Interface Guidelines,* but realize that it's OK to break a few rules for games (such as hiding toolbars and blacking out the screen border during game playing).

6. Leaving out network game support. One of the growth areas in the entertainment business will be network-based games. Take advantage of this revenue opportunity by taking the day or two needed to add AppleTalk, IPX, and direct serial network support to your games. And learn about how the Open Transport system software extension will make it relatively easy for users from different platforms to play together.

7. Not joining an Apple developer program. Joining one of the Apple developer programs is an investment that will pay you back well within the first year of membership in hardware discounts alone. Besides that, game developers benefit from monthly mailings that include informative, time-saving materials that help you more efficiently develop and market your game, such as sample code, development tools, beta versions of new system software, technical support, information about games-related marketing opportunities, and much more.

lation. Though meeting these expectations costs a little more up front, most developers feel that this effort results in a better product for all platform versions.

Apple's Marketing Efforts

Apple's recent shift to a solution-focused marketing approach promises to help game developers, since "entertainment" is one of Apple's five key solution areas. When the dust from this reorganization settles, I think you'll see Apple working even harder to help you get more great entertainment titles on the Macintosh platform, and this means you'll have more opportunities to participate in Apple-sponsored porting kitchens, cooperative advertising programs, and bundling deals.

There are also several marketing efforts under way that are directly helping entertainment software developers. These programs target a problem that many Macintosh developers face—grabbing a fair share of the retail shelf space. And though the number of Macintosh game SKUs (shelf-keeping units) carried by software retailers went up 32 percent last year (source: PC Data in Reston, Virginia), there's still room for improvement. So what is Apple doing to address this problem? Last holiday season, Apple launched a series of United States–based initiatives aimed at helping Macintosh software gain retail shelf space and sales momentum. Apple sponsored no fewer than five retail promotions and four CD-ROM–based catalogs of Apple-related products, providing many developers with product publicity and incremental seasonal sales. (For details, see the article "Apple Initiatives Boost Third-Party Software Sales," on page 21 of the January 1995 issue of *Apple Directions*.) And most recently, Apple began running a series of ads featuring third-party games in a variety of publications that game users read.

A new Apple product that should help expand the market for Macintosh entertainment software is Pippin, Apple's new low-cost CD-ROM playback device derived from Power Macintosh technology. With Pippin you'll be able to move existing Macintosh CD-ROM–based entertainment, education, and multimedia products over to this mass-market platform with very few modifications.

Apple expects an early selling opportunity in the Japanese market as the first Pippin licensee, Bandai Corporation (Japan's largest toymaker and creator of the hit Power Ranger characters) begins shipping its Japanese model later this year. Bandai has publicly stated that it aims to ship 500,000 Pippin devices in the first year. Already, hundreds of Japanese companies—many of whom currently only develop for Sega and Nintendo—have committed to join Apple's Pippin Developer Program. (For more information, see last month's article, "Pippin: A New Platform for Multimedia Titles," on page 17 of the May 1995 issue of *Apple Directions*.)

A Few Words on International Markets

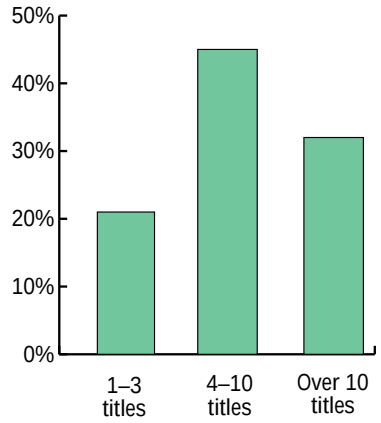
Many game developers have found that there's a strong business case for localizing their entertainment titles for international markets. Outside the United States, the biggest geographic market for entertainment software is Japan, followed by France and the United Kingdom. Two small game companies, Bungie and Graphic Simulations (creator of the F/A-18 Hornet flight simulator), report that Japanese sales total more than 10 percent of their total revenues. And both of these companies reach this market by working through a turnkey Japanese localizer/distributor.

"We're just about to release our localized F/A-18 Hornet product in Japan, and based on advance orders, we estimate that this native version will increase our Japan sales fivefold," says Jeff Morgan, president of Graphic Simulations. "Having a Japanese partner to support your product, localize documentation, and build channel relationships is really important in this market."

Before selling your product outside your own country, you should always test your product with native users. This relatively simple step helps you more accurately forecast sales and avoid sales-limiting cultural oversights.

Entertainment Titles Per Home Computer Owner

(Based on a random telephone survey of 500 U.S. households)

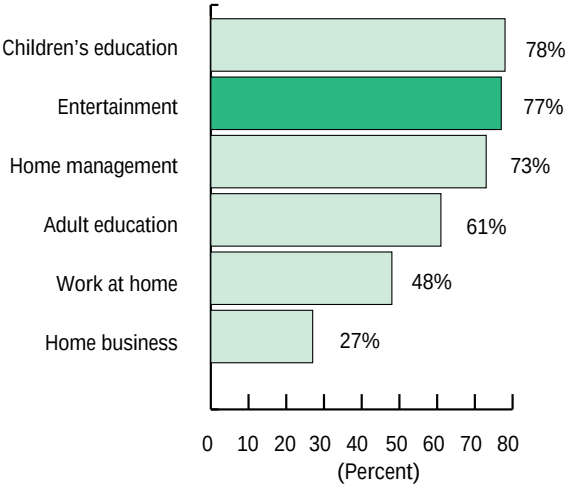


Source: SPA Consumer Report, March 1995

The Entertainment Market Outlook

My outlook for the entertainment market for the next 12 months is optimistic. The installed base of potential game customers will continue to grow dramatically, fueled by the attractive price/performance of Power Macintosh computers. Expect sales of 2 to 2.5 million Power Macintosh computers during the next 12 months, and 5 million in the following year (source: David Nagel, senior vice president of Apple's worldwide

Performa Customer Usage Patterns



Source: Apple Computer, Inc., 1994

R&D organization, at the March 1995 SPA Symposium). And if you haven't begun already, get started on a Power Macintosh "native" game version, because by the 1995 holiday season, Apple predicts that 90 percent of its CPUs will be based on the PowerPC processor.

In the next couple years, I believe the biggest opportunities for game differentiation lie in creating high-quality 3D game experiences and developing games that can be played over networks. QuickTime VR and QuickDraw 3D are cross-platform technologies that will raise the expectations for 3D game experiences. The Open Transport system software extension will provide game developers with easy-to-use industry-standard communication protocols that will enable users on different platforms to play games together.

One of the great things about the Macintosh entertainment market is that it's still a fairly level playing field. The next hit game can come from anywhere, from a Myst-like forest in Washington State to a college dormitory. And with a little luck and hard work, the next hit could come from you. ♣

Special Marketing Report

Recruiting Traditional Animators for Multimedia Projects

By Pamela Kleibrink Thompson

The rapid advancement of computer technology has created an increased demand for high-quality graphics in many types of multimedia products, from computer games to CD-ROM content titles. To achieve this level of quality, multimedia teams can no longer rely on art created by programmers—an artist's touch is needed. But finding artists trained in using computer graphics tools is often a challenge.

One type of artist that's especially hard to find is the computer-savvy animator. Rather than trying to hire these specialists away from competitors (sometimes at inflated salaries), many developers are looking to add traditional animators to their multimedia teams, then train them to use computers. In this article, I share some ideas on how to find, recruit, and integrate these artists into your company.

Why Hire Traditional Animators?

I'm one of those rare individuals who has worked in both traditional animation and software development. As a consultant, I'm currently designing and writing a Honda/Acura interactive training CD-ROM for Gears Communications in Burbank, California, and working as a recruiter for Fox Feature Animation in Phoenix, Arizona.

Previously, I worked as manager of art at Virgin Interactive Entertainment, where I built an art department with people from both the traditional animation industry and the game world. While at Virgin, I was also the supervising producer for the Nintendo version of Disney's *The Jungle Book*. Before Virgin, I worked on animated features such as Paramount's *Bebe's Kids*, the Fox television series *The Simpsons*, and the original episode of *Family Dog* for Amblin Entertainment.

When I speak of "traditional" animators, I'm referring to artists who animate 2D characters on paper for feature films, television, and commercials. To learn their craft, animators have to attend years of specialized training classes. Traditional animation projects are typically organized in an assembly-line fashion, with animators specializing in one or two tasks. Artists may do story boards, character design, animation, cleanup, or ink-and-paint (final outlining and coloring of animations), but each rarely does all of these tasks.

So why should you go to the trouble of finding and training these traditional animators? First and foremost, your customers are familiar with the quality of animation found in television and film, and they'll be more likely to buy your product if you can deliver software products with the same level of quality. The best place to find animators who can achieve this are in the film and television industries. Today there just aren't that many good artists with computer experience, and the few who are working in the computer industry are constantly being recruited from one company to another in an incestuous frenzy. Hiring from the traditional animation industry, you'll be able to choose from a larger talent pool and benefit from less inflated salaries.

Some computer games that have used traditional animation techniques include Freddi Fish and the Case of the Missing Kelp Seeds (Humongous Entertainment); Comic Zone (Sega); BoogerMan (Interplay); Mickey Mania and The Jungle Book (Disney); Lion King (Disney/Virgin); Earthworm Jim (Shing Entertainment); Space Ace (Supterclub); and Brain Dead 13 (ReadySoft).

Perils and Pitfalls

There are, however, perils and pitfalls to hiring traditional animators. At multimedia companies, animators are often expected to participate in all aspects of animation rather than one or two specific areas. If you need a do-it-all animator, make this clear during interviews in order to avoid problems. To some animators the idea of being involved in all facets of the project is appealing. For example, animators who've made their own films have experience in all areas and are potentially better recruits.

Many animators are highly skilled draftspeople who can turn out animation by hand at a rapid pace. But put them in front of a computer, and they slow down dramatically. Many software companies, such as Humongous Entertainment (creators of Putt-Putt and Freddi Fish edutainment products), have evolved an efficient process that combines hand-animation with computer-based work. At Humongous, a professional animator draws the characters by hand; then other people on computers scan in the images, clean up the outlines with a paint program, and fill in colors. To increase productivity, many multimedia companies find that

their studio ends up resembling a traditional animation studio with specialists in animation, scanning and image clean-up, ink-and-paint, and other tasks.

There are also cultural issues that come with hiring a traditional animator into a technical environment: Artists and engineers often have radically different approaches to solving problems and communicating. I address this topic in more detail in the box on this page, "Integrating Artists Into Technical Environments."

And there's one more thing that multimedia developers should be aware of: The Motion Picture Screen Cartoonists Union has established minimum pay scale rates for animators. A beginning or intermediate-level animator can expect to make more than $700 a week; experienced animators make $1,000 a week or more.

Where to Find Animators

There are a number of places to find animators that you may not be aware of if you've never worked in film, television, or advertising. Here are my favorite places to look for "talent."

- *The ASIFA Job Fair.* A great place to start looking for animators is at the International Animated Film Society (ASIFA) Job Fair, an annual event sponsored every spring by ASIFA-Hollywood. With chapters in Chicago, New York, Hollywood, and San Francisco, ASIFA is a networking source open to anyone with an interest in animation.
- *Trade publications.* Place animation job listings in trade publications such as *Daily Variety, The Hollywood Reporter,* and *Animation Magazine.* Other publications that animators frequently read include *Ad Week, Millimeter, Computer Graphics World,* and *Entertainment Employment Journal.*
- *College job fairs.* Many colleges with strong animation programs sponsor open houses and job fairs where you can recruit talented, computer-literate animators into your company. You can also advertise through these colleges' career placement offices and alumni newsletters. Some colleges with excellent animation programs include UCLA, California Institute of the Arts, Rhode Island School of Design, Columbus College of Art and Design, Pratt Institute, New York University, The School of Visual Arts, Columbia, Ringling School of Art and Design, and Sheridan College.
- *Animation schools.* Another good source of animators is the American Animation Institute, which offers classes in animation, story-boarding, and other related skills. Rowland Animation LPV-ROP is a school that also teaches students animation through a regional occupational training program.
- *Word of mouth.* One of the best and least expensive ways to find animators is to use word of mouth. Tell your employees, associates, and everyone you know that you're looking for animation talent. And don't forget to post a notice on online services and the Internet.

Evaluating and Recruiting Animators

In evaluating traditional animators for your company, drawing skills are more important than their knowledge of specific software programs. But make sure the candidate has the ability to learn and adapt, because with the rapidly changing market and emergence of new platforms, flexibility will be key. Remember, it's easier to train an artist to use a computer than to train a computer user to draw well.

In order to screen applicants before you schedule interviews, set up a review board and have your senior artists meet once a week to review portfolios and give recommendations.

Some companies have established tests to determine if an applicant can do the job. This makes it easy to establish if the artist has the necessary skills and can adapt to your style. Try paying promising candidates for a day of freelance work. By having the applicant work side-by-side with your team members, you'll be able to evaluate their productivity and skills, and see if they're a team player. Sure, they'll be on their best behavior, but you'd be surprised at how much you can learn about a person in just one day.

Because of the collaborative nature of animation, a candidate's ability to work in a team environment is important. At Virgin Interactive, many project team members took part in the interview process, in order to help evaluate team dynamics. (If you do this, be sure that all employees receive basic training in interviewing.

Integrating Artists Into Technical Environments

Animation is a collaborative endeavor, whether it's in traditional media, such as feature films and television, or in the computer industry, and good communication skills are essential. In order for a project to come to successful fruition, managers should do everything they can to foster communication between programmers and animation artists. If both respect one another's skills and talents, product development will progress much more smoothly.

Teams of artists and technical people often experience conflict, because of their different communication styles. Artists tend to be more emotional in their expression, while programmers approach things more logically and unemotionally.

Artists bring in new ideas, including character designs and potential licenses, but they're often not technically oriented and have a difficult time adapting to the demands of a technical environment. The time that must be taken to teach artists about file management and other computer basics can often be frustrating to a technically savvy programmer.

Artists often feel reigned in by the technical limitations of the computer platforms they're working on and are occasionally frustrated when they're told by the programmers that their ideas won't work. By its nature, artwork takes up valuable space and memory. RAM allocations for animations, backgrounds, or other graphic elements should be decided upon early in a project's development cycle, and the artists should be kept informed of any changes. As artists gain knowledge of the technical requirements and limitations of the medium, there will be less conflict between the artists and programmers.

In order to successfully integrate artists into a technical team, members of the team must keep the primary goal in mind—that they're all there to make terrific, high-quality software products. All conflicts can be resolved by open discussions and maintaining respect for what each team member brings to a given project.

Make sure they know about illegal questions, such as those about age and marital status.)

Follow rules of good "interview etiquette." Though this may seem too elementary to mention, many great animators have been lost to a competitor because of administrative oversights. Here are my guidelines for treating your interviewees well, because these candidates are potential customers, even if they don't become employees.

- Write up a detailed job description of the type of person you're looking for, including required skills and personality traits. This helps interviewers who might not be familiar with animators.
- Print a schedule of the people who will be interviewing the candidate, including names and titles, and give it to the candidate. Distribute the schedule with the job description and resume of the candidate to all interviewers at least a day before the interview, so they'll have time to prepare.
- When interviewing candidates, respond to them in a reasonable amount of time after each interview. You should update them on their status—for example, if you are still considering them or if you aren't hiring within two weeks of the interview.

One thing to keep in mind if you hire traditional animators is that they come from an entirely different world. You have to sell them not only on the merits of your multimedia business, but also on what makes your particular company special. Tell the candidate why your company is unique. You should be able to communicate the company's mission statement, goals (both short-term and long-term), philosophy, and corporate culture. The candidate should have enough information when leaving the interview to be able to decide in your favor, should you decide to make an offer.

In addition, inform them of recent computer graphics advancements that may make this industry more appealing to them. For example, the move to 64-bit graphics allows for higher-resolution art. Artists may be surprised to learn that they're no longer restricted by computer memory constraints and limited colors.

"Care and Feeding" of Animators

Once you've hired a talented animator, your work has only just begun. To get them up to speed quickly, take the time to orchestrate their integration into your multimedia team.

First, assign a mentor to each new hire, and have that person introduce the new hire to the company and employees. If the mentor was one of the interviewers, that person will take a greater interest in the success of the new hire. Define terms that may be unfamiliar to the new hire, such as *sprites.* Even terms that are familiar to the animator, such as *frames,* may have a different meaning in the world of computers.

Make sure that animators are given the time and the resources to learn software programs. Provide new hires with a copy of each program and its manual.

Traditional animators appreciate a supportive environment, so make them feel that they're an integral part of the development team, and offer feedback on their work. Keep them informed of the project's schedule and status. And make their tasks and goals clear.

Make it a practice to approve animation in stages to minimize rework. And make sure character design is finalized before animation is started. You'll be able to retain animators if they feel they are making progress and are productive.

And finally, be willing to invest in furthering your animator's development. For example, life drawing classes improve their skills.

Before the first day of work, ask new animators if there are any special supplies they need. Most animators use animation desks, animation disks, paper, peg bars, stopwatches, and pencils rather than a computer mouse or keyboard. Most animators' needs can be found at Cartoon Colour, an animation supplies company.

You'll also want to set up a pencil test machine so animators can see if their work flows smoothly or not. A pencil test machine consists of a stand, lights, camera, monitors, and an input device (either a tape recorder or computer).

Some equipment that will make the transition to computers easier includes tablets and stylus (makers include Wacom and Kurta), scanners, and software such as Linker, Amino, AXA, and Softimage. Softimage's Creative Toonz, for example, makes the transition to computers easy since many of the tools traditional animators use are incorporated into the program. AXA Software's product, which was used in creating The Jungle Book and Earthworm Jim, also helps traditional animators make a quick transition.

Tapping Into the Talent Pool

A basic understanding of the process that traditional animators use will help a software title producer find and make the most of new talent entering the field. If you want the beauty of animation in your games but don't want to hire artists, there are many animation houses—such as Nelvana in Toronto, Canada; Character Builders in Worthington, Ohio; and Fox Animation in Phoenix, Arizona—that have experience in CD-ROM titles and video games. Other traditional animation houses, such as Fox Feature Animation and Character Builders, are looking to break into the computer animation business, and would welcome a chance to bid on your animation project.

Tapping into the talent pool of traditional animation can bring new ideas into your multimedia software company and raise the quality (and sales) of your multimedia products. ♣

Pamela Kleibrink Thompson is currently recruiting artists for Fox Feature Animation Studios in Phoenix, Arizona, and The Learning Company in Fremont, California.

Editor's note: You can find an exhaustive list of sources for animators along with the Apple Directions *online posting. See page 3 to find out where you can look for it.*