MACINTOSH

Multimedia

WORKSHOP

Michael D. Murie

Create Multimedia Brilliance

Over 400 M of Multimedia brilliance!

Free CD-ROM
Includes:

Demos
Graphics
QuickTime
Video shorts
Sound Clips
and much more
MACINTOSH MULTIMEDIA WORKSHOP

Hayden Books
MACINTOSH Multimedia WORKSHOP

Michael D. Murie

Hayden Books
Dedication

For Susan...
Credits

Publisher
David Rogelberg

Developmental Editor
Laura Wirthlin

Production Editor
Dave Ciskowski

Editors
Lorna Gentry
Karen Grooms
Pamela Wampler

Publishing Coordinator
Mat Wahlstrom

Cover Designer
Jay Corpus

Interior Designers
Michele Laseau
Kevin Spear

Production Team
Diana Bigham, Katy Bodenmiller, Scott Cook, Tim Cox, Mark Enochs, Howard Jones, Linda Koopman, Tom Loveman, Roger Morgan, Beth Rago, Joe Ramon, Carrie Roth, Greg Simsic
About the Author

Michael D. Murie

Michael Murie is a co-author of Hayden's *QuickTime Handbook*, is the editor of *Exploring Multimedia: The Macintosh Multimedia Newsletter*, and is president of Multimedia Workshop, a Macintosh multimedia consulting firm.

When Michael's not trying to pay the bills by developing animations and interactive programs for clients, or by writing for *MacWEEK* and *New Media* magazines, he works on mad, crazy projects with no commercial appeal. Michael's favorite group is The Beatles: "I always liked the white album because of the diversity of the material, but I can appreciate Sgt. Pepper's place in history."

An acknowledged Macintosh freak, Michael is currently working on the motion picture *The Catnappers*. He has no spare time.
Trademark Acknowledgments

This book mentions too many companies and products to list every trademark on this page. Because we are environmentalists, and because paper is a valuable (and costly) commodity, we limit our trademark acknowledgments to the following statements:

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. Hayden cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Apple, Mac, and Macintosh are registered trademarks of Apple Computer, Inc.
Acknowledgments

As Claude Raines said in Casablanca, "Round up the usual suspects!"
Thanks to all my friends and colleagues for their help and support, both in this book and other ventures.

Dave Ciskowski stepped in at the last minute to see this book through to completion. It wouldn't have happened without his help. Bob Hone, the tech editor, found most of my mistakes (any mistakes that still exist I inserted in the galleys!) and Diana Bigham, Katy Bodenmiller, Scott Cook, Tim Cox, Mark Enochs, Howard Jones, Linda Koopman, Tom Loveman, Roger Morgan, Beth Rago, Joe Ramon, Carrie Roth, and Greg Simsic did heroic work proofing, laying out, and generally making sure the book was turned into a readable form.

I'd particularly like to thank my editor, Laura Wirthlin, for her tireless efforts and understanding. Having a great editor makes an unbearable job bearable.

Particular thanks to the shareware developers who let us include their work on the disc, as well as the representatives from the many companies that provided demos and samples.

Thanks.
We Want To Hear from You

What our readers think of Hayden is crucial to our sense of well-being. If you have any comments, no matter how great or small, we'd appreciate your taking the time to send us a note, fax us a fax, rhyme us a rhyme, etc.

We can be reached at the following address:

David Rogelberg
Hayden Books
11711 North College Avenue
Carmel, Indiana 46032
(800) 428-5331 voice
(800) 448-3804 fax

If this book has changed your life, please write and describe the euphoria you've experienced. Do you have a book idea? Please contact us at the above address.
# Contents

**Part I**  Introduction ................................................................................. 1

1  What is this book? ................................................................................. 3

2  What is multimedia? .............................................................................. 7

    So what is multimedia? ................................................................. 8

    What is multimedia “good for?” ............................................... 9

    Can you do multimedia? ............................................................. 10

    How do you get started? .............................................................. 11

**Part II**  Creating multimedia projects .................................................. 13

3  Developing a multimedia project ....................................................... 15

    The steps of development ......................................................... 16

    Having the idea ........................................................................... 17

    An Example: Electronic Exploring Multimedia newsletter .......... 17

    Establishing the constraints ....................................................... 18

    Obvious Constraints ................................................................... 18

    The not-so-obvious constraints .................................................. 20

    Unique constraints ...................................................................... 24

    Planning the production ............................................................ 26

    The environment ........................................................................ 27

    The data ..................................................................................... 28

    The user interface ...................................................................... 28

    But how to get started ............................................................... 29

    It’s time to start designing .......................................................... 47

    Creating the prototype ................................................................ 50

    Adjusting the prototype ............................................................ 51

    Making the final delivery ............................................................ 52

    All the things we forgot ............................................................. 53

    Moving on .................................................................................. 53

4  Dissecting a multimedia project .......................................................... 55

    Overview of the project ............................................................... 56

    How I became involved ............................................................... 56

    Planning the first prototype ....................................................... 57

    Reviewing the first prototype ..................................................... 59
Switching development tools ........................................ 61
The final phase ............................................................ 62
Producing the final version ............................................ 64
Considering the future ................................................ 66

Part III Multimedia tools .............................................. 67

5 Hardware ................................................................. 69
   A brief history of the Macintosh .................................. 70
   The original Macintosh ............................................. 71
   The color expandable Macintosh II .............................. 72
   The Macintosh LC, or the "pizza box" ......................... 72
   The Portable, the PowerBooks, and the Duos ............. 73
   The Quadras and the Centris .................................. 75
   The Color Classic ................................................. 75
   The Performas ....................................................... 75
   Processor and clock speed .................................... 75
   Choosing the right Macintosh .................................. 76
   The multimedia user's Macintosh .............................. 77
   The multimedia developer's Macintosh ..................... 77
   Considering other hardware .................................. 78
   Graphics display boards and graphics accelerators ...... 78
   Storage devices ..................................................... 79
   Still other hardware ............................................. 84
   The future .......................................................... 84

6 Text ........................................................................... 87
   A brief history of text on the Macintosh ..................... 88
   What text communicates ....................................... 91
   Text ................................................................. 92
   Hypertext .......................................................... 92
   Using text in productions ...................................... 96
   Fonts ................................................................. 97
   Hyperfonts? ....................................................... 99

7 Graphics .................................................................... 101
   A brief history of graphics on the Macintosh .............. 102
   Some tools ........................................................ 108
   Graphics editing programs .................................... 109
Basic object-oriented graphics programs .......... 110
PostScript graphics programs .......................... 111
Graphs or graphics? ........................................ 111
File formats and applications that work with them ........................................ 111
Other sources of graphic images ......................... 117
Video images .................................................. 118
Still video images ............................................. 119
Scanned images ............................................... 120
Kodak Photo CD images ..................................... 122
Clipped images ............................................... 124
Pressure-sensitive drawing tablets ....................... 124
The future ...................................................... 125

8 Animation .................................................. 127
Working with animation ..................................... 129
A few principles of animation ............................. 129
A few problems of animation ............................... 131
File formats ................................................... 133
Some animation tools ........................................ 134
Life Forms ..................................................... 134
PROmotion and ADDmotion ................................. 136
Animation Works ............................................. 138
Cinemation .................................................... 139
Movieworks ................................................... 140
Other animation packages .................................. 141
The future ...................................................... 142

9 3D ......................................................... 143
The three functions of 3D modeling ....................... 144
Basic modeling ............................................... 145
Rendering ..................................................... 151
Animation ...................................................... 160
The categories of 3D applications ......................... 162
Architectural modeling 3D applications .................. 162
Simple extrusion 3D programs ............................. 164
General purpose 3D applications ......................... 165
Considerations when choosing a 3D application .................................................... 166
A look at some 3D applications .............................................................. 167
Ray Dream Designer ................................................................................. 167
Specular International Infini-D ................................................................ 170
Strata Inc’s Strata Vision 3D ................................................................. 171
MacroMind Three-D ................................................................................. 171
Alias Sketch! ............................................................................................. 173
All the rest? ............................................................................................... 174
Hardware requirements ............................................................................ 174
The future ................................................................................................ 175

10 QuickTime ......................................................................................... 177
Understanding the importance of QuickTime ....................................... 179
What’s so neat about digital video? ....................................................... 181
What’s the difference between animation and video? ......................... 182
How does QuickTime work? ................................................................... 183
Installing QuickTime and playing movies ........................................... 184
Recording Video .................................................................................... 186
The compressor’s role in digitizing ....................................................... 187
QuickTime compressors ........................................................................ 189
Some tips for compressing video clips ................................................ 190
Selecting and using digitizing hardware ............................................ 191
Some video digitizing boards ............................................................ 191
Hardware acceleration ........................................................................... 196
Using QuickTime movies ...................................................................... 197
Play them ................................................................................................. 197
Edit them ................................................................................................. 198
Using programs that support QuickTime ............................................ 200
Utilities ................................................................................................. 201
Editing ................................................................................................... 202
Post-processing tools ........................................................................... 204
Other utilities ......................................................................................... 204
Cross-platform applications .................................................................. 205
The future ............................................................................................... 206
### Part IV: Multimedia Environments

#### 13 Choosing the Right Development Environment

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Different Environments</td>
<td>241</td>
</tr>
</tbody>
</table>

---

#### Part III: Sound

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Storing Sound</td>
<td>209</td>
</tr>
<tr>
<td>Macintosh Built-in Sound Hardware</td>
<td>210</td>
</tr>
<tr>
<td>Macintosh Sound Quality</td>
<td>210</td>
</tr>
<tr>
<td>Lower Sample Rates</td>
<td>212</td>
</tr>
<tr>
<td>Sound Compression</td>
<td>213</td>
</tr>
<tr>
<td>Recording Sound</td>
<td>213</td>
</tr>
<tr>
<td>Editing Sound</td>
<td>218</td>
</tr>
<tr>
<td>Saving Sound and Using Sound in Other Applications</td>
<td>219</td>
</tr>
<tr>
<td>Using Advanced Sound Techniques</td>
<td>221</td>
</tr>
<tr>
<td>Getting the Best Sound at 11 kHz</td>
<td>221</td>
</tr>
<tr>
<td>Use a Tape Recorder</td>
<td>221</td>
</tr>
<tr>
<td>Watch Out for Clipping</td>
<td>222</td>
</tr>
<tr>
<td>Avoid Recording Too Low</td>
<td>222</td>
</tr>
<tr>
<td>Get a Good Original for Better Sound</td>
<td>222</td>
</tr>
<tr>
<td>Using Advanced Hardware</td>
<td>223</td>
</tr>
<tr>
<td>Using CD</td>
<td>223</td>
</tr>
<tr>
<td>Using MIDI</td>
<td>224</td>
</tr>
<tr>
<td>Making the Macintosh Speak</td>
<td>226</td>
</tr>
<tr>
<td>Looking Ahead</td>
<td>227</td>
</tr>
</tbody>
</table>

#### Part II: Clip Media and Copyright

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stock Houses</td>
<td>229</td>
</tr>
<tr>
<td>Clip Media</td>
<td>230</td>
</tr>
<tr>
<td>Media Cataloging Utilities</td>
<td>231</td>
</tr>
<tr>
<td>Copyright Laws</td>
<td>234</td>
</tr>
<tr>
<td>Copyright and Your Work</td>
<td>235</td>
</tr>
<tr>
<td>Copyright and Work for Hire</td>
<td>235</td>
</tr>
<tr>
<td>Copyright and the Work of Others</td>
<td>236</td>
</tr>
<tr>
<td>Releases</td>
<td>236</td>
</tr>
<tr>
<td>The Future</td>
<td>237</td>
</tr>
</tbody>
</table>

---

### Part III: Multimedia Environments
14 **Authoring environments** ........................................ 249

HyperCard—what is it? ........................................... 250
How does it work? ................................................... 251
What is it good for? .................................................. 265
Where to go from here? .......................................... 266
The future ................................................................. 267
SuperCard—what is it? ............................................ 268
How does it work? ....................................................... 268
What is it good for? ................................................... 270
Where to go from here? .......................................... 271
The future ................................................................. 271
Director—what is it? ................................................... 271
How does it work? ....................................................... 272
What is it good for? ................................................... 277
Where to go from here? .......................................... 278
Other programs to consider ..................................... 279

15 **Interactive presentations** ..................................... 281

Understanding presentation programs .......................... 282
Using interactive presentations ................................... 283
Using Action! and Special Delivery ............................... 283
Action! ................................................................. 284
Special Delivery ....................................................... 286
The future ................................................................. 288

Part V **The multimedia workshop** ................................. 289

16 **Putting it to work** ................................................. 291

What is the project? .................................................. 292
Getting started ......................................................... 292
Step 1: Setting up the opening screen ........................... 294
Displaying color graphics ........................................... 296
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding a Quit button</td>
<td>298</td>
</tr>
<tr>
<td>Step 2: Adding another background</td>
<td>299</td>
</tr>
<tr>
<td>Add a second background</td>
<td>299</td>
</tr>
<tr>
<td>Add a background text field and buttons</td>
<td>300</td>
</tr>
<tr>
<td>Step 3: Adding content</td>
<td>303</td>
</tr>
<tr>
<td>Add a title</td>
<td>303</td>
</tr>
<tr>
<td>Copying text</td>
<td>305</td>
</tr>
<tr>
<td>Formatting text</td>
<td>306</td>
</tr>
<tr>
<td>Step 4: Adding a picture to a card</td>
<td>307</td>
</tr>
<tr>
<td>Adding a generalized graphic display</td>
<td>308</td>
</tr>
<tr>
<td>What if it doesn’t work?</td>
<td>311</td>
</tr>
<tr>
<td>Finish up the card</td>
<td>312</td>
</tr>
<tr>
<td>Step 5: Adding the other cards</td>
<td>312</td>
</tr>
<tr>
<td>Manually wrapping text</td>
<td>313</td>
</tr>
<tr>
<td>Cards 5 through 10</td>
<td>314</td>
</tr>
<tr>
<td>Step 6: Some finishing touches</td>
<td>316</td>
</tr>
<tr>
<td>Hiding the background field</td>
<td>317</td>
</tr>
<tr>
<td>Adding a button to the first card</td>
<td>317</td>
</tr>
<tr>
<td>Things that haven’t been done</td>
<td>318</td>
</tr>
<tr>
<td>Moving on</td>
<td>320</td>
</tr>
<tr>
<td>Appendix</td>
<td></td>
</tr>
<tr>
<td>What's on the disc</td>
<td>321</td>
</tr>
<tr>
<td>A special note on shareware</td>
<td>322</td>
</tr>
<tr>
<td>Tour the disc</td>
<td>322</td>
</tr>
<tr>
<td>QuickTime</td>
<td>322</td>
</tr>
<tr>
<td>Cover art</td>
<td>323</td>
</tr>
<tr>
<td>Companies</td>
<td>323</td>
</tr>
<tr>
<td>Vividus</td>
<td>323</td>
</tr>
<tr>
<td>Specular International</td>
<td>323</td>
</tr>
<tr>
<td>Playmation</td>
<td>323</td>
</tr>
<tr>
<td>No Hands Software</td>
<td>324</td>
</tr>
<tr>
<td>Virtus</td>
<td>324</td>
</tr>
<tr>
<td>Heizer Software</td>
<td>324</td>
</tr>
<tr>
<td>Fractal Design</td>
<td>324</td>
</tr>
<tr>
<td>Multiple Media Tour</td>
<td>324</td>
</tr>
<tr>
<td>Strata</td>
<td>324</td>
</tr>
</tbody>
</table>
Macintosh Multimedia Workshop

Adobe .............................................................. 325
Movie Works ................................................... 325
Morph ............................................................. 325
PROmotion ....................................................... 325
VideoFusion .................................................... 325
VideoToolKit .................................................... 325
CameraMan ..................................................... 325
Michaels .......................................................... 326
QuickTime Handbook movie .............................. 326
Now & Zen ....................................................... 326
Farmers Market ................................................ 326
Dogcow ............................................................ 326
Exploring Multimedia ....................................... 327
Library ............................................................ 328
Images ............................................................. 328
QuickTime Clips .............................................. 329
Sounds ........................................................... 329
Book chapters .................................................. 329
Index .................................................................. 333
Part I

Introduction
chapter 1

What is this book?
Welcome to the *Macintosh Multimedia Workshop!* The goal of this book is to help those of you who are new to multimedia on the Macintosh get started “doing it yourself.” With the help of this book, you soon will be creating your own multimedia projects.

The first section of the book, “Introduction,” begins with an introduction to the book (you’re standing in it) and then attempts to explain what multimedia is. Maybe you’ve seen this word splashed about the place and wondered what the hype is all about. Chapter 2, “What is multimedia?,” will try to answer all your questions (including some you didn’t even want to ask).

The second section, “Creating multimedia projects,” describes the steps involved in developing multimedia projects. Chapter 3, “Designing a multimedia project,” discusses how to go about designing a project, illustrating the principles with “case studies” of existing projects. Chapter 4, “Dissecting a multimedia project,” provides a retrospective look at a real-world project developed using the steps in Chapter 3. If this section seems like a lot to master immediately, don’t worry—just skip ahead to read about the types of media you can use in multimedia projects and come back to this section later.

The third section, “Multimedia tools,” introduces the different media (text, graphics, animation, 3D, QuickTime, and sound) you can use in multimedia projects. The chapters in this section discuss how you can use and manipulate these media, and introduce some of the applications you might need to start working with them. (True, 3D isn’t really a medium, but it’s such a vast and complicated subject that I didn’t think I could spread it across graphics, animation, and QuickTime without a lot of confusion!) The section begins with Chapter 5, “Hardware,” which describes the equipment you need to run multimedia applications and to develop multimedia projects, and ends with Chapter 12, “Clip media (and copyright),” which provides information about using material created by others.

The fourth section, “Multimedia environments” introduces the environments in which you put together multimedia projects. Chapter 13, “Choosing a multimedia environment,” discusses in broad terms how to do just that. Chapter 14, “Authoring environments,” describes the authoring environments of HyperCard, SuperCard, and
Director in more detail. Chapter 15, "Interactive presentations," describes new developments in interactive presentation environments, including Action! and Special Delivery.

The last section, "The multimedia workshop," shows you how to create a multimedia newsletter from materials included on the accompanying CD-ROM. The newsletter elements provided for this project comprise the inaugural issue of Exploring Multimedia: The Macintosh Multimedia Newsletter. When you finish creating this newsletter, you can apply the same ideas to create a multimedia newsletter or other production of your own.

The CD-ROM included with this book also contains some multimedia illustrations for the book, a variety of utilities, a selection of demonstration applications and sample multimedia projects from companies that sell multimedia products, and some examples of multimedia projects that I have created (most of them for my own amusement!). The appendix describes the contents of the CD-ROM in more detail.

It can be a daunting task to start developing a multimedia project. It is my hope that this book will lead you in the right direction and give you the encouragement to begin developing your own projects. Remember, start small, work a chunk at a time, and keep adding things until you have a complete project. Good luck!

The conventions used in this book were established to help you use the book more easily.

Important terms appear in boldface type where they are defined.

Information that appears onscreen and information that you type appears in a special typeface.

When you see two or more keys separated by a hyphen (-), you hold down the first key(s) as you press the last key.

Multimedia illustrations, utilities, demonstration versions of programs, and sample projects included on the accompanying CD-ROM are designated with this icon. Remember, the appendix describes the contents of the CD-ROM in more detail, and ReadMe files on the CD-ROM may provide more information about specific items.
Chapter 2

What is multimedia?
You could fill several pages of a book with answers to the question: “What is multimedia?” One of my favorite definitions is from a computer salesman, offered when he was setting up hard disks, VCRs, monitors, and other equipment for a demonstration at a local computer user group. When asked what multimedia is, he surveyed the pile of equipment he had just carried into the room and said “Anything that requires more than two trips to the car!”

**So what *is* multimedia?**

Multimedia is often defined simply as the simultaneous use of more than one media type on a computer. That’s a very broad definition, which isn’t precise enough for many. The current use of the term multimedia is particularly curious considering that multimedia used to refer to a slide/sound presentation! Then there’s the issue of interactivity—some users think that you have to have interactivity in your presentation to have multimedia. The truth is that multimedia has been co-opted by the computer industry, and it can be (and has been) used to describe just about anything!

The basic types of media available on a computer are: text, graphics, animation, video, and sound. Another part to many multimedia productions is interactivity—the capability for the user to interact with the multimedia production, causing things to happen or interrupting the flow of information.

Some of you probably are thinking that, by the definition above, your word processor qualifies as a multimedia tool because you can use it to combine two media types—text and graphics. But word processing and even desktop publishing is generally considered an exception to the rule. Why? Perhaps because almost everyone can combine text and graphics, this combination isn’t exotic enough to be deemed “multimedia.” Alternatively, a QuickTime movie, which might contain only animation and sound (or maybe *only* animation!) is considered multimedia.

Gee, only two paragraphs past my definition of multimedia, and it already begins to look like it doesn’t hold water.
In many ways, defining multimedia is like trying to define desktop publishing, but worse. Most people probably would define desktop publishing as the use of a product such as Aldus PageMaker or QuarkXPress to create materials that will be mass produced. Writing a letter with Microsoft Word is not generally considered desktop publishing. But you could use Microsoft Word to create a flyer or brochure or even a newsletter, which might well be considered desktop publishing.

So, writing a letter and placing a graphic in that letter probably does not qualify as creating a multimedia, but writing a letter and attaching a QuickTime movie to that letter probably does!

Maybe a simpler definition of multimedia would begin "If it looks like a duck..."

What is multimedia "good for?"

Perhaps more important than any definition are examples of what you can use multimedia to do. Multimedia is most often used for presentations, training and educational programs, sales demonstrations and museum kiosks, games, and database front-ends.¹ But multimedia also can be used for correspondence, artistic programs, and even wedding invitations! There's really no end to what you can do with multimedia.

If a picture is worth a thousand words, what is a movie worth? Or an animation, or a piece of music? Rather than a text-only explanation of how to assemble a bicycle, why not have a step-by-step animation that shows how all the pieces fit together, and why not enable the user to click on the animated parts to find out their part names and numbers? Similarly, an electronic encyclopedia might use many words to describe the sounds that whales make, without really communicating those sounds. Enabling the readers to click buttons to hear the sounds is a huge benefit. Video is even more powerful—when used correctly—whether as part of an encyclopedia or a sales presentation.

¹What's a front-end? "Front-end" is a term often used to describe an easy-to-use interface for a large collection of data that previously had been accessible only through a clumsy command-line interface.
These possibilities are why multimedia has generated so much excitement, and why you need to know about multimedia and how to create it. Multimedia is not a buzz word for a non-existent technology or a fad that will be around for a couple of years and then fade away. Multimedia exists and is being used for many applications already. The future will see multimedia being integrated into more applications, becoming accepted as part of what a computer “does” rather than some unusual new feature.

Can you do multimedia?

Many multimedia productions available today—particularly commercial games and educational products—would be almost impossible for one person to produce alone. The combination of professional skills needed to produce a high quality production—graphics, animation, video production, sound engineering, and so on—are difficult to find in any one person.

By necessity, multimedia is often a collaborative art form. You might want to specialize in one or two areas (for example, sound and video, or maybe animation or 3D work). Or you might want to be the creative force and guiding light behind a multimedia project (many people do). Having this role is like being the director or producer of a movie, and as director or producer of the project you must bring together its different elements—some which you have created yourself, others which have been created by people you have hired, or have been purchased from clip media collections.

Alternatively, you may be able to produce the whole project yourself. Different needs have different requirements. A commercial product has a very different expectation attached to it than does an internal training product for a large corporation or an educational game for a school. Don’t be frightened by the fact that you might not be able to do it all yourself. In many cases, one person can produce the entire project—particularly if the production involves only one or two media.

I don’t want to scare you into thinking that a multimedia project is some large, difficult development project that is beyond your skills,
time, and budget. Rather, I want to encourage you to start developing your own projects while remaining aware of the possibilities for improving a project by working with others—whether those others are colleagues in your department, friends, or professionals you hire to put the finishing touches on the production.

Designing and producing a multimedia project involves three major tasks, each of which can be done by one person or by several people. The first task is design; this is the process of coming up with the idea for the project and specifying how it will work. The second task is producing the different elements—for example, the art work, the text, and the movies—that you will use in the project. The third task is implementation—taking the elements and putting them together based on the original design.

How do you get started?

Whether you want to design, create, or implement a multimedia project, this book will help you. The next part of this book, “Creating multimedia projects,” describes the creative process—how to go about designing and analyzing a multimedia project. The third part, “Multimedia tools,” describes the tools for creating and manipulating the individual media; the fourth part, “Multimedia environments,” describes how to implement a project. The final part of the book, “The multimedia workshop,” provides an opportunity for you to create part of a multimedia newsletter from elements provided on the accompanying CD-ROM.

Can you do it yourself? Yes! And in the rest of this book, you’re going to learn how.
Creating multimedia projects
chapter 3

Developing a multimedia project
There's more to developing a multimedia project than having an idea, knowing how to use the tools, and putting it all together. You must consider design—graphic elements, such as pictures, and user interface elements, such as buttons and menus—content preparation, and implementation problems. You also must consider whether users will be able to use and get what they need from your product without confusion. All these elements can make the task of creating a multimedia project seem daunting to the beginner.

This chapter provides a step-by-step guide to developing multimedia projects from beginning to end. The chapter also introduces a specific example—an electronic newsletter—the development of which is described in greater depth in Chapter 16. Hopefully, by following this example, you will be able to take the principles and apply them to your own development efforts.

Of course, you also need to know which tools (applications) are available and how to use them. This topic is covered in the two separate sections on Multimedia Tools and Multimedia Environments. Although you can read the chapters in this book in any order, if you are a beginner you may want to read through this chapter and the following chapter quickly first, and then read the sections on the specific applications. Then, when you are ready to actually create your own project, you can reread this chapter again.

The steps of development

Having worked on a number of different projects, I have come up with the following steps in the development of a multimedia project:

1. The idea
2. Establishing the constraints
3. Planning the production
4. Creating the prototype
5. Adjusting the prototype
6. Final delivery of the project

This outline is not some hard and fast formula I have spent years refining. Rather, it's based on an assessment of the steps I usually go through when working on a project. You might work differently, and that's okay. Each project has its own quirks and this outline just consists of the most common steps of development.

Having the idea

Every good project starts with an idea. You have to conceive of some idea, goal, or purpose before you can start working on a project. The clearer the idea, the easier the rest of the development should be. Unfortunately, ideas are often kind of fuzzy—"I want to create a neat product" sums up the goal of many development projects.

An example: Electronic Exploring Multimedia newsletter

For the sample project in this chapter, it seemed like a good idea to develop an electronic version of the newsletter Exploring Multimedia, which I publish when I'm not writing this book. This idea is logical because a) I already have the material and it is an electronic format b) I need something that can be distributed by modem and on disk, c) There's going to be a CD-ROM pressed with this book, which will make for a good way to distribute the newsletter to people who may be interested in reading it, and d) It will be a good example for the book!
It always pays to put the idea into words (written words make the idea seem more real), so here's the idea:

The product is an electronic version of the newsletter *Exploring Multimedia*. The first prototype will use stories from issue 1 of the newsletter.

Pretty simple really. Now that we have the idea, it's time to move on to the next stage.

**Establishing the constraints**

You have probably already started thinking about how this product might work and look. That's great! But before you get carried away with the details, there are some more things you need to know. What are the constraints for the project? Whether you are developing something for yourself, your company, or a client, every project has constraints that will affect the final results. Some constraints are more obvious than others.

**Obvious constraints**

The most obvious constraints are cost, time, and materials.

**Cost**

Cost is usually the biggest constraint. How much money is available for the project? Money, of course, frequently influences how much time and effort you can put into creating a project. Limited funds may mean that you can't afford a professional sound track or narration. Or maybe you can't afford the cost of licensing software or distributing several floppy disks to every user. These are all issues that will affect the final design.
For the electronic newsletter I am paying for the development, so cost is a serious issue. Luckily, the major expense is my time, so as long as I don't overpay myself I should come in within budget!

**Time**

When is the project due? Whether developing a project for a client or for yourself, you will have some kind of deadline. Even if you have "unlimited" funds, you may not have the time to do everything you have planned.

This project is a prototype version of the newsletter—I will only implement one article and part of the structure as an experiment. If the experiment works out, then I will implement the rest of the issue; otherwise, I might try some other method for distributing the newsletter electronically. Since it's a prototype, the finished piece was produced in less than a week.

**Materials**

Multimedia developers need to have the right equipment and the right software available. If you don't have the right equipment, you may not be able to develop the project at all. For example, if you don't have a color-capable machine, you won't be able to make QuickTime movies. The right software also can mean the difference between a successful and an unsuccessful project. You should refer to the chapters in the section on Multimedia Tools to establish the hardware and software that you may need for your project.

Software and hardware can be really hairy for the beginner. Often you know what you want to do, but you don't know what you need to do it. If you want to create some illustrations, should you buy Illustrator, Freehand, Photoshop, Painter, Canvas, or some other program? Often you won't know until you buy the program and invest some time in learning it.

1 Sometimes clients will tell you they have unlimited funds. Usually that means they have a very tight schedule and they know there is no way you can outspend their budget in the limited amount of time!
To help determine which programs are appropriate for which tasks, read the chapters on tools in this book, and read reviews in Macintosh magazines. If you can at least identify what you need to do, you will narrow the field down to only a few programs.

The cost, time, and materials constraints are, in a sense, the macro-constraints. Multimedia projects live or die by the micro-constraints—the things that affect the end-user, rather than the problems the developer had when putting the product together.

The not-so-obvious constraints

These constraints include such things as: Who will use the product? What will they use it on? What do they expect from the product? You might think that these things are obvious, and simply skip to the next section. However, before spending a great amount of effort creating a sophisticated product, you should be careful to make sure that all your users will derive the most benefit from it!

Who will use the product?

It's important to know who is going to use the product and what they will expect from it. In reality, very few products are "for everyone." Even programs such as spreadsheets and word processors are for particular segments of the computer-user market. An exhibit at a museum or an electronic sales brochure will be used by a very different audience than an electronic help system. Each group of users has different needs and expectations.

The user environment might have different requirements, too. This is especially true for programs running in public places. A museum exhibit or a shop demo may require a limitation on the maximum amount of time a user will spend using the system. For example, the average user probably will spend no more than five minutes using the product. To achieve this goal, you may have to cut the material so that there is only so much for a user to see.
Applications for public installations also often require some kind of automatic reset and attract mode. After a period of non-use, the product will go back to the beginning and run some animation in an attempt to attract someone's attention.

The example project: The electronic version of Exploring Multimedia is primarily for people who are working with multimedia tools, or who are seriously considering doing so. It's therefore unlikely that the casual Macintosh user will use this product. We will assume (it is dangerous to assume, but at some point you have to) that these serious users will want solid information. Although they may be impressed by (or may even expect) a sophisticated-looking product, these users don't want a lot of fluff.

What will they use it on?

What kind of computer will these users have access to? This question is important. If a large proportion of the users will only have a Mac Plus or SE, then they comprise a very different audience than Quadra users. Hardware restrictions often limit what you can do. You can, of course, create multiple versions of the product—for example, one for color and another for black-and-white systems. Or you can have the product itself determine what kind of machine it is running on, and then adjust accordingly. Developing for this kind of situation will, in most cases, increase production costs because of the additional time required to create and test multiple versions of graphics. If you have two versions of the production, one for color and one for black and white, then this increases the amount of disk space, or requires that an inventory of two versions be maintained. Also, fixing problems in two versions takes longer (and inevitably causes problems).

Some other examples of hardware limitations are:

QuickTime

At this writing, QuickTime runs only on computers that support Color QuickDraw. None of the 68000-based Macs meet this requirement, so Mac Plus, SE (except for the SE/30), and PowerBook 100 users are unable to play QuickTime movies. If your product requires QuickTime, then those users will be unable to play it.
Creating multimedia projects

Color

Color animation or graphics will not always work effectively on black-and-white computers. At best, color graphics may appear murky; at worst, they may display as a solid black square. Is this a problem? Of course it is a problem, but is it a problem for the vast majority of users? If most of them will have color machines, then you might just ignore the problem and require that the user have a color monitor.

What version of the System?

System 7 added color icons and a new desktop structure. If you want to make use of color icons, aliases, and other System 7 features, don't forget that users running System 6 will not see these things. System 7 has added many other new features that more applications are taking advantage of. Although today only a few applications will not run under System 6, this could become more of a problem in the future; you really need to consider what version of the System your users will be running.

How much memory?

Memory became a concern with the arrival of System 7—which takes up even more memory than System 6 did. Most multimedia productions require a lot of memory, and it is not unusual to require a minimum of four or possibly eight megabytes of memory to run a presentation. At the same time, it is perfectly possible to produce something that will run in two megabytes (the minimum for System 7), and enable most Macintosh owners to run your product.

Variations in playback speed

A fantastic animation created on a Quadra 800 may run so slowly on a Mac II that the animation loses its effect. QuickTime alleviates some of these problems (or attempts to), but the reality is that you will always be able to create something that looks spectacular on the fastest, hottest machine—and abysmal on the slowest and lowliest. Always check your work on the slowest machine recommended for the production.
Screen size and bit depth

In the early days, every Macintosh had the same size screen—512 x 342 pixels. Then came the Mac II, and for a long time the standard color screen size was 640 x 480 pixels. When the LC was released, Apple unveiled a 12-inch color monitor with a pixel resolution of 512 x 384. Things have only gotten worse—the color PowerBook 165c has a completely different pixel resolution of 640 x 400.

If you create a presentation that completely fills a 640 x 480 screen, what happens when you play it on a smaller screen? In most cases, the rest of the image will be cropped. Unfortunately, most multimedia authoring tools do not allow you to change the size of the presentation. For this reason, it is probably best to use the smallest screen size that your audience is likely to have.

Pixel depth—the number of colors possible on screen—is important. It’s often safe to assume that color monitors have 8-bit pixel depths (and can display 256 colors at one time), but 24-bit screen depth cannot be assumed. Also, you probably should avoid 24-bit screen depth because animations and graphics become very large and animations play slower when they are in 24-bit color (three times the amount of information is being moved around, after all).

Tip:
The safest solution is to only work in 1-bit (black and white), and always optimize for playback on a non-System 7 Mac Plus. If you do that, then you know that anyone who owns a Macintosh will be able to play and see your production. However, sometimes you have to make tradeoffs. For each case, you have to ask “Is this a problem?” and “Can we live with the outcome?” Which is more important, a state-of-the-art presentation that blows the doors off the competition, or something that the widest audience can view? For different applications, the answer will be different.
How will the product be delivered?

This is an important question that is sometimes overlooked. How will you get the product to the end user? For one-off presentation situations like a presentation to a potential client, you can use a SyQuest cartridge, DAT tape, and even a portable hard disk, and then hand-deliver and install the product.

If you are sending copies to multiple locations, then you have to consider what equipment the users have access to, and how much it will cost to deliver on a given medium. For example, although many users may have SyQuest drives, the cost of the cartridges makes these drives unattractive for mass distribution (see also Chapter 5, "Hardware").

Even using floppy disks is a problem. Can your users read high-density disks? Although high-density (HD) disks hold 1.4M of "stuff" and make distribution much easier, not everyone has those drives, so you may have to go with 800K, double-density disks instead.

It doesn't take many disks to make distribution on CD-ROM cost competitive. If you are distributing several hundred copies, then a production that takes up more than two or three floppy disks will cost less to distribute on CD-ROM. Of course, not everyone has CD-ROM drives. Do you want to limit yourself to those users?

Unique constraints

Some constraints may be unique to your application. To determine all the constraints, create a list for each of the categories like the one in Table 3.1.

Table 3.1 Unique project constraints

<table>
<thead>
<tr>
<th>Element</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Development Authoring tools</td>
<td>What tools will you need to use? Do you have the skills to use them? Do you have to learn, or hire someone else? Any licensing requirements? Programs such as HyperCard and Passport Producer</td>
</tr>
</tbody>
</table>
### Developing a multimedia project

<table>
<thead>
<tr>
<th>Element</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware</td>
<td>Have licensing requirements, whereas others (SuperCard and Director) do not. What equipment do you have? What might you need? This equipment might include video or sound-digitizing equipment, or a hard disk for mastering a CD.</td>
</tr>
<tr>
<td>Data</td>
<td>In what format is the information used in the product? If the project uses a large amount of data from different sources, how is the data formatted and how will it be accessed by you?</td>
</tr>
<tr>
<td>Resources</td>
<td>What other resources—for graphics, video, sound, or data formatting—will you need? What will it cost?</td>
</tr>
<tr>
<td>Production</td>
<td>Will the production or the data need to be updated? How will the production be updated? Who will do it, and what is involved?</td>
</tr>
<tr>
<td>Updates</td>
<td>You might design and build the product but some of the information may be added by someone working for the client. Who will do this work? Will they need special instruction? Special software?</td>
</tr>
<tr>
<td>Implementation</td>
<td>How much material do you need to distribute? What equipment do users have? What is cost effective?</td>
</tr>
<tr>
<td>Distribution</td>
<td>How and who will duplicate and distribute the production? What are the costs?</td>
</tr>
<tr>
<td>Media</td>
<td>What hardware do users require to run the production? What hardware do most users have?</td>
</tr>
<tr>
<td>Duplication</td>
<td>How will users use the production? Will they need any additional help (manuals, written instruction)? Will users require any other support—where will they get it?</td>
</tr>
</tbody>
</table>

At this point, you should be able to define a minimum configuration. The minimum configuration is the minimum amount of equipment a user will need to use or play your production.
Creating multimedia projects

For the electronic newsletter, it's probably safe to assume that the complete version should be smaller than 750K when compressed so that a) it fits on a floppy disk and b) it doesn't take too long to download using a modem. I'm also going to assume that users have a color-compatible Macintosh. This is a safe assumption because the audience for the newsletter is people using Macintosh multimedia tools, and most of these people will be using color at some point.

This is the minimum configuration for the electronic newsletter:

- System 6.0.7+
- Color-compatible Macintosh
- 256 color monitor or better
- Screen resolution 512 x 342
- Color Classic²

Planning the production

Creating a multimedia project is difficult because everything is dependent upon everything else. The choice of tools and authoring environment, the users' requirements, and the material or subject matter of the production all limit the design, and the three work together to make the job of creating a design more difficult.

Working on your first project is particularly difficult because you don't fully understand the capabilities of your tools or the structure of the data, and no one ever understands the requirements of the user. These three elements are interrelated, and while you can consider them separately, they also must be considered together. Often it is only at the very end of a project that you understand the project well enough to actually create a good plan for implementing it. And by then it's too late! All you can do is hope that you get to produce a similar project and apply some of what you learned to that.
Let’s look at the three elements that make up a production in depth. They are:

- The authoring environment
- The data
- The user interface

### The environment

The environment is the program you use to create your multimedia production and bring all the elements together. This program should not be confused with tools such as paint programs, QuickTime editing programs, and other programs you use when creating your production.

Director, HyperCard, Passport Producer, and Special Delivery are all authoring environments. They are described in greater depth in the section on multimedia environments. If you want to learn about them, read Section IV, “Multimedia environments.”

This section assumes that you are not going to program your production from scratch using a programming language such as Pascal or C++. If you are (and it costs a lot of money and time to do so), then ignore this section and skip to the next one.

Whichever authoring tool you choose, you will be limited in the things you can do with the tool. There is no perfect multimedia environment—just many applications with different strengths and weaknesses. For example, Director is very good to use for animation, but it is much more complicated to script, while HyperCard is good for scripting but only supports a single window, and SuperCard... well, you get the idea.

The authoring environment you choose will affect the look and functionality of the final production. Some applications can be used to create almost anything, while others may have more limited uses. If you are starting out and trying to decide which program to use (particularly if you are planning to spend time learning the application), then choosing the right program can be daunting. To make a
decision, play with other productions created with the program you are considering. Try and find something with a similar look and feel to what you plan to do. And read the section on multimedia environments (Section IV).

The data

The data is the heart of the production. It's the information you are trying to convey to the user. In the electronic newsletter, it's the articles in the newsletter. For a sales presentation, it's the sales figures and product specifications.

When you begin designing your production you have to ask yourself: What am I trying to provide to the user? An online help system is very different from a sales demonstration, which is different from an interactive game. Each has its own characteristics. An online help system may consist of a large amount of information that the user needs to search in a number of different ways. A sales demonstration is usually linear and may run from beginning to end without user interaction. An interactive game may have a linear story that is divided into segments; each segment may consist of many branching choices that the user explores while trying to progress to the next step in the story.

The user interface

It is very important to understand what the data is, and how the user will interact with it. Your first task is to make sure that you fully understand the data. This task can be particularly difficult when you are presented with a new project you have to design an interface for.

There are two kinds of elements in user interface design: structural and cosmetic. Both are important, but you need to understand their differences. Structural elements include items such as menus, data fields, windows, and buttons. Cosmetic elements include items such as background graphics and the shape of icons.
Structural elements must be defined early in development, because changing these elements involves large changes in a project. Cosmetic elements are usually easy to change, and you can add these elements towards the end of the project. Many elements of a design have both a cosmetic and structural part to them. For example, an icon in a window that accesses a Find command is a structural element; however, the image of the icon is cosmetic. Adding the Find command at the end of the project could be very difficult, but changing the shape of the icon will probably not be.

One of the problems of multimedia is that it is a very visual (some would say superficial) medium, and although the structural elements are the most important, too much time is spent worrying about the cosmetic details. As a multimedia developer, you have to learn how to devote an equal amount of time to both, and not be overly distracted by one or the other.

**But how to get started?**

Getting started is difficult, but take heart. "Immature artists imitate. Mature artists steal."\(^3\) Why spend months reinventing the wheel? Spend some time examining other projects that are similar to yours, and then take the elements you like the best and use them in your project!

---

Note that under copyright law ideas are not protected, only the expression of the idea.

For the electronic newsletter, I looked through several multimedia projects Apple has produced, as well as some projects from other developers. Apple spends a lot of time and money on their products, so you know you are working with the best when you "borrow" ideas from them!
The Macintosh OnLine Reference

The Macintosh OnLine Reference is an electronic collection of tips and hints for Macintosh users, including everything from 'How To Turn Off the Trash Warning' to 'Setting the Alarm Clock.' This project was created in HyperCard.

Figure 3.1. The structure of the Macintosh OnLine Reference. See the text for a description of the functions of each screen.
The structure of the Macintosh Online Reference

Basically, the Reference is a collection of tips presented on one or more HyperCard cards like card 4 in figure 3.1. Although this information is the heart of the system, there is a whole structure that provides different ways for a user to find the information that he or she needs. This includes an index, table of contents, and a Find command.

The Main screen, screen 1, consists of three buttons (top left), a graphic (right), and a brief welcoming description (bottom left.) The three buttons provide access to a Main Topics section (essentially a table of contents), an index (an alphabetical list of key words), and a Look For command (a Find command that lists all occurrences of a word.)

Clicking on the Main Topics button in screen 1 accesses a list of Main Topics (screen 2). Clicking on one of these topics (bottom left of the screen) displays a list of subtopics to the right of the topics, as shown in screen 3. Selecting a subtopic (by clicking on it) takes you to screen 4, an actual topic information screen.

Note that in screen 3 you can click on a subtopic (the list on the right), choose another topic (the list on the left), or even access the index or the Look For command by clicking on the buttons (top left.) All these options remain available until you choose an actual topic, at which point the information screen appears (screen 4). From the information screen, you either can return to the list of topics (screen 3) by clicking on the List of Topics button at the bottom of the window, or you can browse through the information cards using the left and right arrows (in the lower right corner). Two other buttons
In screen 5, clicking on the Index button displays an alphabetical index below the buttons. You can scroll through this list, or you can click on a letter of the alphabet (above the field) to jump to that part of the index. Clicking on an entry in the index brings up a sublist of the index (screen 6.). Note that the index, although it contains different information, works in a manner similar to the Topics list; the major items appear in a scrolling list on the left side of the screen, and the subelements appear on the right side of the screen. Clicking on an item in the subindex on screen 6 takes you to the information screen (screen 4).

Clicking on the Look For command button brings up a field that asks you to enter the word enable you to print the screen or access the Glossary (described later). To use the index or the Look For command, you must click on the List of Topics button to go back to screen 3.
for which you are looking. Clicking on the Get Topics button brings up a list of matches on the right side of screen (7). Clicking on one of these topics takes you to the information screen (screen 4).

The Macintosh OnLine Reference includes some other information contained in a glossary. These items are available in a separate window (actually a separate HyperCard Stack) called the Glossary. The Glossary is available from the menu (not shown here) or from the Glossary button that appears on the information screen (screen 4). The Glossary appears as a list of Glossary Terms in a smaller window (screen 8). This list works in the same way as the list in the index; it’s a scrolling list with an alphabetical jump-to-letter control at the top of the list.

When you click on an item in the Glossary list, the entry and its definition appear (screen 9). Notice the Speaker button next to the title of the entry; when you click on the Speaker button, the word is spoken. Some of the words in the definition are underlined. Clicking on an
underlined word takes you to the entry in the Glossary for that word (screen 10), and so on. You can return to the Glossary list by clicking on the List of Glossary Terms button at the bottom of the window.

When you return to the Welcome screen (screen 11), note that the Welcome text suggests a couple of places to go to get more information about how to use the system. Both of these options are available from a menu; choosing one of the options accesses the window shown in screen 11. The first screen in the window is an overview of the reference, while the other screens (not shown) describe how to use it.

Evaluating the Macintosh OnLine Reference

The Macintosh Reference is an interesting example of an electronic information system. However, before we either blindly copy the ideas or look at something else, it pays to spend some time trying to evaluate why it was designed the way it was. By doing that, we will learn from what Apple did and be able to apply what we learn to our own projects.

- The Table of Contents and Index make the product similar in function to a book.

- There are some nice computer-like features added to the book model. These include the capability to jump quickly from one place to another, the Look For command, and the speaking glossary. The Glossary is linked to itself (key words are marked in the definitions), and the information cards also contain links to the Glossary.

- The Table of Contents, Index, and Look For command all work in a similar way. After users have tried one function, they should have no problem using another.
• The Welcome screen—the first screen the user sees—immediately tells the new user what to do to find out how to use the system, but the user who just wants to experiment can immediately do so.

• Graphics have been merged with the text in the information cards to provide a very pleasant experience. However, note that there is no animation (although it could be argued that animation isn’t needed for this particular reference guide).

• Although the two-level system (topic, subtopic) works well for this application, it may not work well for other systems. (What if you have three levels—topic, subtopic, sub-subtopic?)

• Although the electronic reference provides a lot of the features found in regular books, it lacks others. First, it’s almost impossible to tell how big the book is! When you look at the information cards, you have no idea how many other cards there are (although it does tell how many cards there are for the current topic (top right of screen ④)). When you pick up a book, you know how big it is, and therefore have a rough idea of the amount of information it contains (and the amount of time it would take to read it). This is not so for most electronic books. For a reference source, this is not so much of a problem (it’s unlikely that users will read it from cover to cover), but it may be a problem in other situations.

• Another problem is a lack of bookmarks. You can’t mark somewhere that you have been so that you quickly can get back to it. It would be great (and fairly easy) to add this feature.

All in all, the Macintosh OnLine Reference is a very good example of a multimedia production, and definitely worth looking at.

QuickTime Intro News

When Apple introduced QuickTime, it also pressed a CD-ROM called the Apple CD-ROM Titles Sampler. This CD-ROM contains two multimedia projects: the QuickTime Intro News and the Titles Sampler (both described below). Graphically and functionally, these two products are different; however, in many respects they perform a
similar task—both contain a directory of products available from different companies.

QuickTime Intro News runs in HyperCard, just like the Macintosh OnLine Reference; however, visually it is a very different production (see figure 3.2). All the screens have color graphics (displayed using the ColorizeHC XCMD; see Chapter 14, "Authoring environments") and a sophisticated look with color backgrounds, animations, and other graphic items. QuickTime is used to play animations as well as small video interviews.

Figure 3.2. The structure of QuickTime Intro News. See the text for a description of the functions of each screen.
The structure of the QuickTime Intro News

When you first run the QuickTime Intro News, an animation is played (screen 1) that displays the title and looks very interesting (the first time you see it). When the animation finishes, the Intro News screen appears (screen 2). In some ways, this resembles the first page of a newspaper, with the masthead at the top of the screen along with the date of publication. Then there are two “articles” that begin on this screen and continue elsewhere (indicated by the two buttons labeled Continued...). Clicking on one of these buttons accesses screen 3 which continues the story. Note that screen 3 also contains a QuickTime movie (top left) that you can play.

Notice the Index button at the bottom of each screen. Clicking on that button takes you to the index screen (screen 4). You will notice something odd about this index—it’s not an index of articles; instead, it’s a list of manufacturers. Intro News contains a directory of different products that support QuickTime. To see the information cards,
each card contains two or three entries. The entries themselves contain a short text description, and many have a QuickTime movie you can play.

The table of contents (screen 6) is available by pressing the button at the bottom of the screen. The table of contents screen contains several elements. The two buttons top left point to the two stories that start on the first page—so clicking on either of the buttons will take you back to the first screen. The third element (top right) takes you to another screen (screen 7, described below). The other elements below these three buttons take you to the master categories in the stack.

Screen 7 contains the other article in this publication. This article consists of a short description of the QuickTime Starter Kit (a product that Apple sells) along with an illustration showing you what the QuickTime Starter Kit contains. What you can’t see in this illustration is that the screen first appears blank, and then a box appears onscreen. The various elements of the Starter Kit “fly” out of the box and arrange themselves on the screen. It’s a very entertaining animation.

**Evaluating the QuickTime Intro News**

The QuickTime Intro News is very different from the Macintosh OnLine Reference in structure, form, and appearance. It’s almost hard to believe that the same company produced them both. The products
were designed to serve different functions—the Reference provides information to a user, while the Intro News seems to be more of a sales tool.

The lessons to learn from this project are:

- Visually, this electronic publication is very appealing, but...

- When you run this product, you are not exactly sure of what you are dealing with—is it a newspaper or a directory?

- Greater emphasis should have been placed on the table of contents so that users would be able to understand the structure of the production. Also, the two parts (the articles and the directory) should have been more clearly separated.

- More work could have been done to make the directory information useful to the user (much of this was done in the second production, the Titles Sampler, which is described below). Perhaps the Titles Sampler, which is a separate HyperCard stack with its own interface and information, resulted in the developers limiting the functionality of this product.

- Although the animations are entertaining the first time, they are slow and repetitive after that. Keep this point in mind for your own projects. You might want to store a parameter so that a “cute” animation only plays once (or give the user the option of turning off the animation).

To conclude, the QuickTime Intro News is an interesting mix of personalities that can’t quite decide what it is. The color, graphical design, and integration of QuickTime make Intro News worth a look, but you should be able to improve upon the basic structure for your design.

**Titles Sampler**

The Titles Sampler (see figure 3.3) also is included on the Apple CD-ROM Titles Sampler. Just like the Intro News, the Titles Sampler is created in HyperCard, and has a color interface. But the graphic design and the functionality are very different. The Titles Sampler is a directory of products that can be accessed in a number of ways.
Figure 3.3. The structure of the Titles Sampler. See the text for a description of the functions of each screen.
The structure of the Titles Sampler

The Titles Sampler begins with animation and a title screen (screen 1). You must click on the screen to go to the next screen, which is titled View By (screen 2). From this screen, you have three options—Title, Compatibility, and Topic. Clicking on the Title button accesses a list of titles (screen 3). You can scroll through this list, or use the alphabet on the right side of the screen to jump to a letter in the alphabet.

Although the topics in the list appear in alphabetical order according to product title, it is also possible to list the products according to publisher. Simply click on the View by Publisher button at the top of the screen; the screen changes to screen 6.

When you click on a title (either in the By Publisher or By Title list), the product information screen (screen 4) appears. This screen includes an image from the product and a short text description. A button under the image will run a demo (if a demo is included on the disk).
Creating multimedia projects

In the lower left corner of the screen, you will see the Add and Show buttons. The Titles Sampler can keep a list of the products in which you're interested. When you click on the Add button, the product is added to the list; when you click on the Show button, the list is displayed.

Clicking on the Ordering Info button on screen 4 (to the right of the Add and Show buttons) takes you to the ordering info screen (screen 5). This screen shows the system requirements for the product, as well as the ordering information. To find the ordering information for different locations around the world, you click on the world map. When a different location is clicked, the ordering information in the ordering info field (on the left side of the screen) is updated.

In addition to using the Titles list, you also can access the list of products through a Compatibility list and a Topic list. You select these lists on screen 2, as you do the Titles list. Both of these lists (screens 7 and 8) are similar to the Titles list, but the row of buttons on the right side of the screen changes to reflect either
a list of compatibility requirements, such as CD-ROM or Color, or topic categories, such as Business or Desktop Publishing.

Evaluating the Titles Sampler

The Titles Sampler is much easier to understand than the Intro News. The Titles Sampler contains only one type of information, and it is pretty clear to the user what that information is, after only a little exploration. The small number of categories is a limitation, and when you first look at an information card, it is not clear what the product is unless you read the description card. Is it hardware or software? Is it a paint program or a game? Also, the Find command is very rudimentary.

Multiple Media Tour

The Multiple Media Tour, published by the Audio Visual Group, is a CD-ROM containing a large collection of multimedia resources (images, sounds, and so on). Included on the Multiple Media Tour disc is an electronic catalog, which makes for an interesting design example (see figure 3.4).

Figure 3.4. The structure of the Multiple Media Tour. See the text for a description of the functions of each screen.
Created in SuperCard, the Multiple Media Tour is a directory of all the multimedia resources on the CD-ROM. The table of contents screen (screen 1) is the first screen you see when you run the program. This screen provides a list of categories. Clicking on a category (in this example, Textures) takes you to a subcategory screen (screen 2). This screen divides the textures into several sub-subcategories. Clicking on one of these sub-subcategories takes you to a screen showing all the materials in that sub-subcategory (screen 3). When you click on one of these materials, the screen shows the actual image itself (not shown). You then can copy the material into the Clipboard and paste it into another program.

Not shown is a navigation palette that contains a button you click to return to the table of contents, and buttons to move through the cards in the stack.

The Multiple Media Tour is a very simple interface that works for this particular application. It operates in a logical fashion that anybody should be able to use. There are a few minor problems (for example, the Help button is on the category screen, but not on any of the other
screens), but these problems don’t really interfere with the operation of the product.

Macintosh Human Interface Guidelines Companion

The Macintosh Human Interface Guidelines Companion is another interactive online reference from Apple. Unlike the Macintosh OnLine Reference, this product is in color and was created in SuperCard. (It's much easier to add color in SuperCard than in HyperCard.)

Figure 3.5. The structure of the Macintosh Human Interface Guidelines Companion. See the text for a description of the functions of each screen.
Like the Macintosh OnLine Reference, the Human Interface Guidelines basically consists of a series of cards with information about different topics (screen 5). Because these cards are in color, they are more graphically appealing than the cards in the Macintosh OnLine Reference. However, the real difference between the products is in the method used to access these cards.

As in the Macintosh OnLine Reference, you are presented with a table of contents that starts with the major headings (screen 1). Clicking on a topic accesses subtopics (screen 2). This process continues down two more levels (screens 3 and 4) until you reach the information screen (screen 5).

Visually, the Human Interface Guidelines is much more appealing as well as clear—you always know where you are. You can click on any of the parent categories to jump to another topic.

On the down side, although this design works well for this particular product, it will be difficult to add more than one more level to this screen—although it's unlikely that you will have to deal with that many subtopics in a project any way.

As an implementation criticism (rather than a design criticism), the choice of the 640 x 480 screen for this product means that users with smaller screens will not be able to view it easily. However, Apple obviously intends this product to be used by developers, and developers will more than likely have larger color screens.
It's time to start designing

Because multimedia authoring tools are so easy to use, it's tempting to start prototyping right away—especially with an application such as HyperCard or Director. There is nothing wrong with this approach; however, I recommend that you first create a storyboard, ideally on paper (see figure 3.6).

There are several reasons for using a storyboard. First, it will concentrate your thinking. I have nothing against building prototypes and then improving them—it's the way I prefer to work—but it's important to start with an idea of the direction in which you are heading. Second, if you are working for a client, a storyboard provides an opportunity for the client to correct all your misunderstandings before you spend countless hours working on the project.

You can create your storyboard in a number of ways. At first, you might just sketch things on paper—it's still a lot faster than drawing with any program available for the Macintosh! After you have an idea of what the project looks like, you can use an object-oriented drawing program such as Canvas or MacDraw to flesh out the layout. These programs are preferable to bit-mapped programs such as Photoshop or MacPaint; you will find it much easier to make changes in object-oriented drawing programs. I like to create layouts in PageMaker because I can easily combine graphics and text in the storyboard.
On the left of figure 3.6, blocks represent the screens in a project; on the right, text describes the contents of the screen and anything a reader might need to know about how the project works. Note that interactivity—how you move from one screen to another—is indicated by lines going from one screen to another.

**Figure 3.6.** A storyboard.

After the storyboard is completed, review it with the client, users, and anyone else you can find. The more input, the better. Also, by this stage, you should have chosen an authoring environment. The authoring environment will have a direct impact on the next phase.

The next step is to consider any interface issues that relate to specific screens (for example, how buttons work, what's contained in fields, and so forth). Depending on your project, this process may require very little or a lot of thought and work. If the project is reasonably simple, and you are using design concepts used in other
applications, then you might be able to skip to the next task immediately. Also at this point, you can start work on some of the cosmetic issues, such as what graphics you need, where will they come from, and what format the data is in.

For the electronic newsletter, I created a number of sketches and ended up with the diagram in figure 3.7.

![Sketch for the electronic Exploring Multimedia newsletter.](image)

Figure 3.7. Sketch for the electronic Exploring Multimedia newsletter.

There were several false starts in the design process. I originally thought that MacroMind Director was the tool to use because it supports QuickTime and many animation effects. As the sketches were produced and I studied the information that the newsletter contained more closely, it became clear that text handling was going to be important. Unfortunately, Director is not as good with text as it is with animation.

SuperCard would have been a good choice, because it supports color. It doesn’t support QuickTime, but for the prototype that would not be a problem. A problem that was unexpected was that the publisher wanted readers of this book to be able to experiment and build the prototype. That meant that we had to use the development tool that most readers would have access to. So HyperCard was the final choice—using XCMD’s to provide color. It’s not the ideal solution, but it’s the one we will have to go with (at least for the prototype).

A lot of time was spent trying to figure out how the article would hang together—would it be spread over several cards, or would the text be in one scrolling field? If the latter example was used, then where would the figures that accompany the article be stored?
The final design consisted of:

- A master card from which each article could be accessed.
- Each article is contained on a sequence of cards. Buttons let the user move forward and backward through the article.
- A dictionary would be attached that would contain references and all the occurrences of a word in the newsletter.

Other things that might be considered are:

- How are other issues distributed? Are they added to existing copies or are they separate?
- Let the user add bookmarks and keep notes.
- Author, subject and title indexes.
- Search for any text.

To see how the prototype turned out, read Chapter 16, “Putting it to work.”

Creating the prototype

Now the fun begins—creating the prototype. One developer has suggested that the best way to create a great product is to engineer a prototype, throw away the complete prototype, and build the final product from scratch! Although this sounds like a great way to produce a product, few have the time or resources to work this way.

As an alternative, the best way to work is in “chunks.” This first part of the project will be implementation of the structure—the fields and most of the buttons. These elements are the things that everything hangs from, and it makes no sense to start designing graphics before the structure is complete. After the structure is in place, try and implement specific functions. This method is the best way to work with large projects. Choose functions that can be tested independent of the rest of the project. Test to see what the users think, and then change or add to the prototype.
Why not build the whole thing? If you like to work that way, then certainly do so. But for large projects, or if you are not sure what you are doing, you will be better off if you implement part of the project and see how that goes.

One of the great advantages of authoring tools such as HyperCard and Director is that they make it so easy to change and add things. This is both a strength and weakness. After you have changed something several times—particularly scripts—the scripts can become very complicated and inefficient because you change bits and pieces instead of recreating the whole thing from scratch.

## Adjusting the prototype

Whether you create just a part of a prototype or the whole thing in one pass, at some point, you will want to show the prototype to others for input and suggestions.

It’s a difficult thing to show your baby to others—particularly a prototype that doesn’t have all the fancy graphics and sound that you imagine it will have. Much to your frustration, many people will start picking apart the very things you haven’t finished yet—"Why aren’t there any graphics?" and "Why doesn’t that button highlight?"

Unfortunately, there’s no way around it: you have to show your prototype to others, and you will just have to deal with what they have to say. Very few people know how to give criticism—many people don’t know how to criticize the work rather than the person ("Why on earth did you do it in that color?"). Try not to take it personally.

Try to get all the input you can, from as many sources as you can. One of the best ways to test an interface is to show what you have to someone who has never seen the project before. Try not to tell the potential user how to do anything. Resist the temptation! Just sit and watch what the person does.

Developers with a lot of money often hire independent research firms to set up tests that are videotaped. Videotaping has two advantages: you don’t have to be in the same room and you can’t interfere and say things such as, "Click the top button!" Also, you can view the videotape over and over again.
You don’t have to incur the expense of videotaping if you can’t afford it, but you will have to work hard to learn how to sit and watch what users do without trying to help. Do they know how to find things? Can they use the index? Do they seem to find everything in the system?

After testing, adjust the prototype. Make any changes, and then test it again. You might have to change something several times before it works. If part of the interface doesn’t work, make sure that you get another new user to test it when you make a change. If you show the project to original users, they already will have an idea of how the old way was supposed to work, and that may help them with the new version.

Be prepared to work through several iterations until the project is complete. However, be careful not to lose sight of the objective—it’s all too easy to keep going round and round in circles trying to get something just right.

It’s also important to keep track of any problems in the prototype that need to be fixed. Keep a list of things you need to fix, and check them as you do them. Keep a second list of all the things that are done to the project and when. This is particularly important during the final stages of development. The first list is a list of bugs, while the second list is a list of bug fixes as well as any new features or other changes you decided to make.

**Making the final delivery**

The final delivery is the completion of the project. No matter how stimulating a project is, it’s always great to finish something and send it out into the world. For the final delivery, you will have to consider some things you may not have worried about for the prototype, such as documentation and installation. If you want to include documentation, you can provide it on paper or electronically as balloon help or a “read me file.” If there are any special installation instructions or processes for the project, include those as well.
All the things we forgot

You shipped the final delivery to the client and two weeks later they call you up to tell you that they need some changes made. Don’t be surprised—it always happens! Clients will report bugs that weren’t found during testing, and new features that need to be added after the project is “completed.”

But it’s also important to spend some time looking at the project dispassionately to learn for future projects. Let some time go by, and then go back to the project and try it out. Does it work the way you expected it to? What would you change if you had to do it over? What can you change to make it better? You may never go back and make those changes to this project, but maybe on the next one you will put all you learned to good use.

Moving on

In this chapter, you looked at the development process in depth, both in theory and for a demo system. In the next chapter, you can review the development process for a particular project.
Chapter 4

Dissecting a multimedia project
The previous chapter, "Designing a multimedia project," led you through the steps for creating a multimedia project. Hopefully, that chapter will help you organize your project, and give you the courage to get started.

Because the design process still may seem abstract, this chapter explains how a real multimedia project was created. First, a little background information is in order.

Overview of the project

MarketPlace Business is a CD-ROM database that contains over 7 million U.S. businesses; it is used to produce mailing lists and cold call lists for marketers. MarketPlace users select prospects using criteria such as type of business and sales revenue. Then MarketPlace generates a list of names that match the criteria. The information is encoded on the disc, but the user can purchase names as she wants them; once names are purchased, the user is free to use them for almost any purpose.

MarketPlace uses HyperCard as the front end—the graphical user interface—to the database. HyperCard enabled the developers to produce a graphic, simple-to-use interface. MarketPlace also had a multimedia help system that features animated "movies" that show how to use the system. This chapter describes how the movies were developed, and some of the issues encountered during their development.1

How I became involved

When Lotus began developing MarketPlace, Rob Lippincott, head of the Multimedia Department at Lotus, was busy trying to promote the use of multimedia within Lotus. Rob realized that the MarketPlace CD-ROM would provide the space to add multimedia help for users. He advocated and promoted the use of multimedia in MarketPlace with the MarketPlace project team, and used outside contractors to develop some early HyperCard prototypes.
When I joined the project, the prototype design and interface for MarketPlace had been completed. Using the prototype interface design, a multimedia prototype called “Dave goes Prospecting” had been created in HyperCard. This prototype consisted of a sequence of cards making up a “movie.” The movie introduced Dave, who wanted to use MarketPlace to find new prospects for his business. The movie showed some screens from the product intercut with pictures of Dave.

The movie had an audio narration, but was very limited in structure; after the movie started, it could not be stopped or paused, and had to play to the end. The animation in the movie also was very limited. Rob felt that MacroMind Director would improve the quality of the animation and offer features that had not been possible with HyperCard alone. That’s where I came in. I was hired to use Director to develop a prototype that expanded upon the ideas in the original prototype. The final prototype was a result of many people’s input, including Rob Lippincott, Marcia Zuckerman, Cindy Null, Lars Jensen, Marian Stern, and Andy Hoillinger.

Planning the first prototype

Before building a new prototype, Rob and I reviewed the current one and decided what improvements to add. We felt that some means of controlling the movie should be available for the user—something like a VCR control panel. Also, although the narration worked well, we were concerned that it might be inaudible in loud environments, so we decided to add captions. Finally, the first prototype had featured one long story about Dave and how he used the product. We decided to try a different approach: a story with several subparts.

In our new approach, each subpart would demonstrate a particular function of MarketPlace. The help system would consist of one story about Janet trying to accomplish a task using MarketPlace. The movie would explain what Janet was trying to do and then show how she solved the problem using a feature in MarketPlace. The movie would consist of several tasks, each illustrated by accompanying “how to” explanations. A user would be able to view the whole movie, or, if he...
needed help on a particular topic, the user could view only the segment of the movie that showed how to solve that problem. We referred to this new approach for the first prototype as the Pearls & Swine structure (see figure 4.1).

![Diagram of the Pearls and Swine prototype structure.](image)

**Figure 4.1.** Pearls and Swine prototype structure. The entire movie is made up of segments. The "How to" segments, called Pearls, could be played separately to find out how to perform an individual task in MarketPlace.

We eventually ended up with the prototype in figure 4.2. (This is really the fourth version of the prototype, but it is perhaps the best example of the ideas we had.) The prototype featured an audio narration, captions, and animation, as well as a control panel that allowed the user to stop, rewind, and fast forward.

![Image of the first prototype.](image)

**Figure 4.2.** The first prototype.
The movie consisted of parts showing Janet and parts showing the product. We even had Janet appear with the screen of the product. This technique was an attempt at an "agent," and was useful for indicating specific MarketPlace features. For example, Janet would point to where the user should look when the narration said, "Click the Export Button...".

**Reviewing the first prototype**

In figure 4.2, notice that the actual product screen has been shrunk to make room for the control panel. This change also made it clear that you were looking at a demo, not the real product.

Along the bottom of the screen, you see the control panel. The first icon represented the whole movie. Pressing that button played the movie from beginning to end. The next four icons represented the "How To" parts of the movie. Pressing one of these buttons played only that segment of the movie. The icons in the lower right corner enabled the user to go to the beginning of the movie, rewind, pause, play, or fast forward. Above these controls, you see the exit button—a three-pronged spear (the code name of the product was Trident).

At this point, the prototype was reviewed by the MarketPlace documentation team. The movies would be part of the help system (which included paper documentation and an electronic online manual that was separate from the movies). The other team members had been involved peripherally in the first prototype because they provided the script for the movie, but up until this point the other members had not seen the prototype.

The team decided to abandon the idea of shrinking the product screen. The small screen was too hard to read, and users would have difficulty following the program. Each screen also had to be hand-tweaked to make it readable—a tedious process.

By this time, we were beginning to question the Pearls & Swine idea. We weren't sure if the scripts were the problem (because we didn't really know how to write a script in this new format) or if the idea itself was wrong.
Creating multimedia projects

The next prototype featured a full-size screen and a completely different control panel (see figure 4.3). The control panel was now a palette that only appeared onscreen when the user paused the movie. The icons representing the segments still were displayed in the control panel. Also, in an attempt to make the whole project more cohesive, we cut back on the animation and the screens that contained story segments about Janet. As in the previous prototype, the basic concept remained the same: the movie contained segments that could be viewed together or on their own.

![Figure 4.3. The second version of the prototype.](image)

Because the control panel was not visible while the movie was playing (it obstructed too much of the screen), a small box containing the words Click at any time to stop the movie and bring up the controller was displayed while the movie was playing. We spent quite a bit of time experimenting with the location of this message and the control panel so that they wouldn’t obscure anything important on the product screen.

After this prototype was completed, the team made two decisions. First, we abandoned the idea of segmenting movies that could be watched as a whole. We were unable to make the idea work. The movie was too long and disjointed, and the individual segments weren’t in-depth enough. We decided to split the movies into two types: a dozen segments that explained each part of the product, as well as three movies that were introductions to the product and showed someone performing a task with MarketPlace.
Second, we gave up on the idea of Janet acting as an agent. We just couldn’t get the concept to work. Whether Janet was on top of the screen or inside a little window, her presence distracted too much from what was going on, and didn’t add anything to the experience. We filed the idea away for another project.

**Switching development tools**

The first two prototypes had been developed in MacroMind Director, and we had planned to use the MacroMind Director Player (a set of XCMDS) which plays Director animations within a HyperCard stack. Remember, HyperCard was being used to develop the rest of the product. But we were beginning to have doubts about using MacroMind Director. At the time we began development, HyperCard 1.2.5 was available; we knew that HyperCard 2.0 would be released soon, and the product had to work with both versions. But we had no idea whether the Director Player would work with HyperCard 2.0.

There were also questions about the Player’s stability when running under MultiFinder (which was required to run MarketPlace) and the amount of memory that the XCMD and animation required when running.

We also discovered that many of the animation effects that Director offered were no longer in use. The early prototypes, for example, contained buttons that grew to 400% of their actual size to draw the user’s attention. But this animation had been removed; it was becoming clear that we could do nearly all of the animation effects using HyperCard. At this point, we decided to abandon Director and use HyperCard exclusively to make the movies.

The one animation requirement we did have was to simulate in HyperCard the cursor moving across the screen. In our first attempt to simulate this movement, we used a special font that contained a character like the HyperCard hand. This character was placed in a transparent field on the screen, and the position of the field was changed rapidly to animate the field and simulate the cursor moving around the screen. This method (except with a button instead of a field) was used in the final version, and it worked quite well. In fact,
on faster computers it worked too well, and we had to add a timing routine to slow down the animation effect on faster computers.

Because of the way font characters are displayed, at first we reversed the cursor. This was done for two reasons. With this technique, the cursor did not appear transparent—you couldn’t see what was under the cursor. We also thought that reversing the cursor would lessen the chance of the user confusing the “real” cursor with the imaginary one—but more on that later.

The final phase

We continued with the control panel, but to make the job of hiding it easier, the control panel was now a field containing font characters (see also Chapter 7, “Text”). The prototype had almost reached its final form (see figure 4.4), but there were still many problems to solve.

![Figure 4.4. The final design for MarketPlace help movies.](image)

In the first prototypes, we experimented with animation effects to attract the user’s attention to what was happening onscreen. We created buttons that grew to mammoth proportions so that the user couldn’t miss them. We created sliding buttons, blinking cursors, and several kinds of call-out boxes. We even had Janet point at things.

The problem was that although we wanted to attract the user’s attention, we wanted to do it in a subtle way—but not in such a subtle way that the user would be surprised when the actual product didn’t
perform exactly as it had in the demo. We didn’t want users saying, “What happened to that box that said ‘Click Me’?” After a couple of different attempts, we finally resolved the issue of how to let the user know where to look on the demo screen. We used highlight boxes to point to areas of the screen. These boxes drew the user’s attention without interfering with the appearance of the product.

At this point in development, we also were concerned about the amount of disc space the demo would occupy. Although a CD-ROM appears to be almost endless, we discovered that the actual data could easily fill the disc. Each of our movies was taking up 2 megabytes of space, and we had been allocated only 20 megabytes (and the project managers were keen to get any of that back).

More testing followed. Lotus hired a user interface consultant to test the interface of the product, and he also did some testing of the help system. These tests involved putting people who had never used the product before in front of a computer and asking them to try to use a copy of the prototype. While the new users worked with the prototype, their actions were monitored. Any problems were noted and reported back to the development team.

During this trial period, we discovered our biggest problem was that users confused the movie with the actual product. We wrestled with this problem constantly. How could we make the user aware that it was a movie and not the actual product? (I should add that some of us thought the movie should play inside the product, and if the user wanted to actually use the product, then that was fine.) One possible solution was to play the whirring noise of a film projector before the movie started, but would this be too cute and would a younger audience be familiar with the analogy?

Users also sometimes responded to the narration as if they were working with the real product. When the narration said “Now click the business icon...,” some users reached for the mouse and tried to do just that. So we changed the narration to be less directive. The narration now said something like “To enter the business options, you would click the business icon.”

The issue of terminology in the narration was very important. Do you “click,” “select,” or “hit” items onscreen? Luckily, as part of the online documentation effort, terminology had already been standardized, and this was used in the movie’s narration.
Another problem was that users weren't always aware that they
could actually stop the movie. The "Click at any time to stop the
movie" message now had been moved to the top of the screen
because we couldn't permanently cover up any other part of the
screen with the message. The top of the screen wasn't an ideal place;
if captions were turned on, the message was hidden. We also discov-
ered that people never looked there; they were too busy following the
animation. When the user did notice the message (usually when it
was pointed out to them), the word "stop" suggested to most users
that clicking would stop the movie and take them back to the pro-
duct. Users were too scared to stop the movie because they thought
that they wouldn't see the end of the movie and would have to sit
through the beginning all over again. So we changed the wording to
"Click at any time to pause the movie."

Towards the end of the prototype process, we realized that the
reversed cursor was not acceptable (for an aesthetic reason mostly),
so we had to go back to a plain cursor. This process involved switch-
ing to an animated button rather than a field. Still the debate about
the cursor raged. Some wanted the "real" cursor hidden while the
movie was playing to reduce confusion. Others argued that the user
would be confused if the real cursor weren't visible! And, unfortu-
nately, we didn't have time for more testing.

Figure 4.5. Three ways to emulate the HyperCard cursor.

Producing the final version

After the basic structure and a prototype of the movies had been
designed and completed, we began writing the scripts and producing
the animation. This process was reasonably straightforward.
The documentation team wrote the scripts and also checked the scripts. To ensure accuracy, the scripts were circulated to other project managers for review. Then, the script writer and movie "producer" read each script in front of a computer, while performing the appropriate functions described in the script. This step helped us find things that should be highlighted in the movies, and it also provided a thorough last check of the scripts—it was amazing how many minor errors were found in the scripts even after many reviews.

A scratch audio track was recorded using MacRecorder. Each movie was assembled using screen dumps from the product. Some of the screens were improvised because the user interface was not finalized (or even working). Several people reviewed these movies and any errors were fixed. After this first pass, several of the scripts were changed to reflect changes in the product's user interface.

Because we developed the prototype while the user interface of the product was still incomplete, we had to go back and make a lot of changes, which increased the cost of the project. Unfortunately, this was the only way we could get the help system completed in time. The situation was aggravated by the last minute changes the software developers made to the user interface of the product—these changes meant we had to change all our documentation!

The movies were changed or remade to reflect the final product changes, and we hired a professional narrator to record the final sound track. The sound track was recorded at a sound studio, and then digitized from tape using a MacRecorder. We used a sampling frequency of 11KHz, which provided a satisfactory quality playing on the Macintosh internal speaker without taking up too much disk space.

Although the movies themselves were only three or four megabytes in total, the sound files took up about 18 megabytes. We discovered, just as we put the whole prototype together, that HyperCard could not handle more than 16 megabytes of resources in a single stack. The sounds had to be taken out of the master stack and placed in two resource files that were opened by a special XCMD.

Just before we finished the job, Farallon released MediaTracks, a product that can record and play back actions performed on a computer. Although it came out too late, we considered using
Creating multimedia projects

MediaTracks instead of HyperCard. I think MediaTracks would have saved some time, but it would have required more time to implement captions and the control panel, and it would not have helped us when we were developing our prototypes. MediaTracks is no longer widely marketed, but there are some new products that capture screen actions to QuickTime movies—see the section on CameraMan and Spectator in the QuickTime chapter.

Considering the future

Although I'm pleased with the prototype we ended up with, I can't help but think about the future. I'm haunted by one prototype we explored, which we dubbed "Obi-Wan." Obi-Wan was a concept for a help system that would be available at all times. Its control panel would always list items the user was most likely to need help with at that moment. To provide a comprehensive help, Obi-Wan would link together demonstration movies, an online user guide, and point-and-click information about anything onscreen. Unfortunately, we didn't have the time or resources to fully explore that prototype, but maybe next time!

The next section of this book describes the tools for creating and manipulating different media types. The first chapter discusses special hardware that you might need when working with multimedia.
part III
Multimedia tools
This chapter is about the hardware that multimedia developers and users need for creating and running multimedia productions. This chapter does not cover all of the hardware you will use—hardware issues relating to specific media topics are covered in the appropriate chapters. Read the “Graphics,” “3D,” “QuickTime,” and “Sound” chapters for discussions of hardware that relate specifically to those topics.

Perhaps the question asked most often about hardware is “What Macintosh should I buy?” That used to be a fairly easy question to answer—back when only three or four Macintosh models were available, and new models were released only once every eighteen months.

Those were the good old days—pre-1991! Since then, Apple has been releasing new hardware at a faster and faster clip. Add to that the dizzying array of add-on products and it’s easy to become confused about what to get.

Still, it doesn’t help those of you thinking about making a purchase to hear that things are just getting more confusing. So, the first section of this chapter provides a brief history of the Macintosh and the second section provides a general guideline for buying a Macintosh. Even if you already own a Macintosh, you should read the second section of the book, which includes descriptions of other hardware you might need or want to buy. The final section looks briefly into the future.

A brief history of the Macintosh

In the beginning computers were designed for manipulating numbers. After a while, they were used for text handling—database applications and then word processing. Then came graphics and sound, and now computers are being used for video. This evolution of capabilities has made multimedia possible.

The Macintosh has also evolved over the years, though it’s interesting to note just how capable the original design was. When
introduced in 1984, the original Macintosh had 128K (kilobytes) of memory, a single 400K floppy disk drive, and a 68000 processor. In 1991, Apple was selling the Macintosh Classic, the case had been redesigned, the motherboard had been updated, 2M (megabytes) had become the minimum memory configuration, and a 1.4M high density floppy disk drive was standard, but the processor was still the same 68000. And the basic capabilities of the operating system—text, graphics, and sound, were still the same. It’s hard to believe that Apple was able to sell a computer based on the same processor for so long, and it really is an indication of what a good job Apple did with the design of their operating system—or maybe how little progress there was in the rest of the computer field.

The original Macintosh

The original Macintosh had a built-in black-and-white display with 512 x 346 pixels. It had built-in sound support—a speaker with hardware to play 8-bit, 22 kHz (kilohertz) sound. It also had a set of standard graphic routines (called QuickDraw) that applications used to draw on the screen. This configuration represented a dramatic difference from the PC world, where different graphic display cards required different routines to draw graphics. Macintosh graphics and sound capabilities are more fully explained in Chapter 7, “Graphics,” and Chapter 11, “Sound.”

The original 128K Macintosh was followed by the 512K Macintosh, and then the Macintosh Plus. The Plus had 1M of memory, an 800K floppy disk drive and a SCSI (Small Computer System Interface) interface, which made it relatively easy to attach external hard disks and other devices such as scanners to the computer.

The Macintosh Plus had many advantages, but it also had many disadvantages. It did not support color, and it was not expandable—you couldn’t add plug-in boards to digitize video or display color graphics on a separate monitor. The Mac Plus evolved into the SE, and then the Mac Classic. Both machines were functionally similar to the Plus.

It’s important to remember that the Plus, SE and Classic exist in large numbers. If you are creating a multimedia presentation for mass
distribution, you will need to develop something that will run on these models. They support sound, 1-bit graphics and animation. They do not support color, and cannot currently run QuickTime movies, though this may change (see Chapter 10, “QuickTime”).

The color, expandable Macintosh II

In 1986 Apple released the Macintosh II line, which supported color and had six slots for plugging in boards that used the NuBus standard—a standard for plug-in cards—Apple had adopted.

The Macintosh II used the 68020 processor and had new ROMs that included the code for supporting color. Over time came the Macintosh IIX (which had a 68030 processor), the Macintosh IICX (which was a IIX with only three slots), and the Macintosh IIIi (which was a IICX with a faster processor). The Macintosh IIIfx had the same basic shape as the Macintosh II and IIX, but had a much faster processor. The IIIfx also had its share of compatibility problems, but that’s another story. The Macintosh IIIi was the last model in the Macintosh II line. It had only one NuBus slot, which made it cheaper than the other Mac II models.

The addition of color and plug-in cards made the Macintosh much more multimedia-friendly. Plug-in cards made it possible to add video digitizers and high-quality sound boards, while color is necessary for video applications, image retouching, or just to display a photographic image on the screen. The 68020 processor was much faster than the 68000 processor used in the Plus. This extra horsepower was necessary when working with the larger files made possible with color, among other things.

The Macintosh LC, or the “pizza box”

The Macintosh LC strikes a line midway between the original Macintosh Plus base and the Macintosh II line, which was expensive
because of the options it offered. Apple figured that by stripping out the NuBus slots, but providing color, it could offer a much cheaper computer to home and school users. The original Macintosh LC wasn’t as popular as was expected, but was followed by the LC II and then the LC III, which is almost as fast as the IIci. An added attraction of the LC III is that an FPU (Floating Point Unit) can be added to the computer, making it possible to use the LC III with 3D applications (see chapter 9 “3D”).

The Macintosh LC has sometimes been described as a “pizza box” because of its flat shape. Unfortunately, the LC does not accept NuBus cards, though a PDS (processor direct slot) makes it possible to add specially designed cards. For example, SuperMac makes a version of the VideoSpigot video digitizing card that plugs into the LC.

While the LC might not be a great authoring machine, it does offer color and can play QuickTime movies, and is cheaper than the larger Macintosh models. It is primarily a good player machine.

The Portable, the PowerBooks, and the Duos

When Apple finally announced a portable version of the Macintosh, there was much excitement among the ranks of Macintosh users. Then they saw the Portable. Weighing about a ton and a half and larger than a VW, it wasn’t the kind of thing you wanted to be carrying around much.

The original Macintosh Portable used the same processor as the Macintosh Plus (though running at a faster clock rate) and had all the features of a Macintosh (though no color or slots). Perhaps because it was so large, it didn’t prove very popular.

Apple got things right when it released a new range of portable Macs called the PowerBooks. Originally offered were the PowerBook 100, 140, and 170. The 100 had a 68000 processor and no color ROMs. It was smaller than the other PowerBook models because the floppy disk drive was separate from the unit. The 140 and 170 were similar in shape, and both were larger and heavier than the 100. The
PowerBook 170 had a better screen and a faster processor than the 140. These models have since been replaced by the 145, 160, and 180 models.

For multimedia developers the biggest difference between the 100 and the other PowerBook models was that the 100 could not run QuickTime because it lacked the color ROMs. For other reasons, the PowerBook 100 was not the success that Apple had expected, and it was discontinued.

The PowerBook makes a very good portable machine for playing multimedia applications. The latest models have ports on the back for plugging in external color monitors, making it possible to show color presentations if there is a monitor available at the location. Alternatively you can simply use the PowerBook's screen. The PowerBook line now includes a color model (the 165c). As a development platform, the PowerBooks' biggest drawback is the trackball—most people find it very difficult to manipulate buttons and so forth using the trackball. You can solve that problem by using an external mouse.

In some ways the Duos replaced the PowerBook 100 because they are as small and light as the 100. But in reality the Duos are much more powerful—they have much faster processors than the 100, and they support QuickTime. When plugged into a base unit, the Duos turn into a desktop computer, with a floppy disk drive, NuBus cards, and all the characteristics of a Macintosh II-class computer. Used this way, the Duo might be a better authoring solution than the PowerBooks.

The Duos and PowerBooks do share one minor annoyance for multimedia users. Both product lines, because they run on batteries, feature power-saving features which slow down the speed of the processor when the operating system thinks that the computer isn’t doing anything useful. Unfortunately, sometimes when an animation is playing, the operating system might erroneously think that the user isn’t doing anything, because the user hasn’t clicked or typed anything for a few minutes.

There’s a couple of solutions to this problem. One is to move the mouse during the presentation—a little tedious but it works. The other is to go into the PowerBook control panel and turn off the power-saving option when playing the presentation (refer to the manual that came with the PowerBook to find out how to do so).
The Quadras and the Centrises

Apple's current "top of the line" is the Quadra series. The Quadra 950 and 800 are the fastest computers that Apple sells. If you use digital video applications such as Digital Film from SuperMac or MoviePak from RasterOps, then you probably need a Quadra.

The Centri s line has replaced the mid-range Macintosh II models. The Centris models are excellent computers for multimedia authoring jobs, because they are fast and are much cheaper than the previous comparable models. If you are planning to use 3D modeling, however, make sure the model you buy has an FPU (Floating Point Unit).

The Color Classic

The Color Classic, released in early 1993, added a color screen and a faster processor to what had been the basic Macintosh shape. Although it does not provide a great multimedia authoring environment, this computer can be very attractive for interactive installations because of its small size, all-in-one configuration, and comparatively low price.

The Performas

In 1992 Apple released the Performa line. These are relabellings of existing models (the Macintosh Classic II, LC II, and Macintosh IIvx), and are sold through mass distribution outlets such as Sears. For playing multimedia these machines are very good, but as an authoring environment only the most powerful of the first Performa models, the 600 (Macintosh IIvx) is of interest.

Processor and clock speed

One factor in choosing a computer is the speed of the machine. A Macintosh Plus is much slower than a Quadra because of the speed of the processor. Existing Macintoshes use either the 68000 (Plus etc.), 68020 (Macintosh II), 68030 (I Ice and most PowerBooks) or
68040 (Centris and Quadras). The higher the number of the processor, the faster they work—this is generally because the larger processors can work with larger amounts of data. Another variable is the clock rate. This is the speed at which the processor works with information. The higher the clock rate, the more work the processor can perform in a given amount of time. A 68030 running at 33 MHz (megahertz) will be much faster than a 16MHz 68030. (But a 25MHz 68040 will be faster than either processor.)

Which chip do you need? The faster the chip, the faster the machine, and that’s a good thing. But more importantly, the operating system routines that support color will not run on a 68000 machine—that’s why you don’t want a Plus if you want to work with color, or if you want to play QuickTime movies. If you want to work with 3D applications, then the existence of a FPU is just as important as whether the machine has a 68030 or 68040 chip.

**Choosing the right Macintosh**

Choosing the right Macintosh can be difficult, particularly when you are just starting out. Often you don’t really know what you want to do with the computer, which makes choosing the right one even more difficult. There is, however, one rule that has remained true so far:

*It will be cheaper next year.*

Whatever computer you buy, it will be cheaper and faster next year. In fact, the computer you buy today very easily could be obsolete tomorrow. If you can, put off buying a computer for as long as possible. This doesn’t mean, however, that you should postpone working with computers. Sure, if you wait, the equipment will be cheaper, but in the meantime you are missing out on opportunities, and others are advancing. So, if you can beg, borrow, or steal computer time from other sources, then postpone making a purchase. But at some point, you will have to jump in and take a chance.
The multimedia user's Macintosh

Although it would be nice if a multimedia user could buy any Macintosh and be able to run any available multimedia production, this is unfortunately not the case. Many current multimedia productions have very specific requirements:

- color
- 13-inch or larger monitor
- 5M or more of memory

If you have a Macintosh that meets all these requirements, then the chances are very good that any multimedia production can run on your computer. But, just in case it can't, be sure to check the requirements of a multimedia product before making a purchase. If the Color Classic, with its smaller color screen, becomes popular, then you can expect to see multimedia developers adjust to that size as a standard.

Of course, these requirements do not necessarily apply to special multimedia productions. For example, if a museum is planning to build and install an interactive system, it can specify the minimum requirements and build its multimedia presentation to match those.

The multimedia developer's Macintosh

Do not despair that the constant changes in hardware make it impossible to keep up with the latest hardware available. Just as you can't afford to keep buying new equipment, neither can anyone else!

For many multimedia developers the Centris 650 or equivalent will be the best purchase because it is fast and has slots for plugging in video digitizing cards. The Centris 610 is also interesting, but it has only one NuBus slot—and the slot is not long enough for all NuBus...
cards. That can be a serious limitation. Also, the 610 does not currently have an FPU, which makes it unsuitable for 3D applications.

The Macintosh LC III is almost as fast as a lici (the previous "best" multimedia authoring computer). The LC III, however, has only a PDS slot (not a NuBus slot), which makes it more limited than the Centris 610.

If you have the money, the Quadra series offers the most power, but at the highest cost. Only if you know you really need the speed and expansion options of the Quadra line should you consider buying one of those models.

**Considering other hardware**

After you have purchased your Macintosh, you might think that you have all you need to start working. Sadly, this is not the case, and you may need to consider buying several other products to improve or augment the performance of your base computer.

**Graphics display boards and graphics accelerators**

All the Macintosh models available have some type of built-in video support. You still may need to buy a separate video graphics card, however. Third-party cards offer several benefits, including support for alternative bit depths and built-in graphics acceleration.

For example, the Quadra 800 built-in video card displays only 16-bit color, but you could buy a 24-bit color card and use that if you need to view graphics in 24 bits. What bit depth do you require? For photographic retouching work, you probably will want 24 bits. For video work, 16 bits is often preferred because the quality of the image is
similar to 24 bits, but the data rate is reduced. The bit depth, of course, is primarily a consideration when creating the video. You can display a 16-bit movie on a 24-bit screen and vice versa, but if you view only 16-bit movies on a 24-bit monitor, you aren’t going to notice any improvement, so why pay extra? For 2D animation work, 8 bits is often all that you need. For most 3D rendering, 24-bit color will let you see the best-looking images, though you don’t have to have a 24-bit display to render the images.

Storage devices

Multimedia productions, particularly those that involve sound or QuickTime (video), quickly fill hard disks and any other storage medium you might have available. Unfortunately, there is no hard and fast formula for predicting how large your productions will be. You should buy the biggest hard disk drive you can afford, with 300M as a bare minimum. (This number does vary depending upon the other types of storage you might have available. If you have a removable disk of some kind, then you might be able to work with less hard disk space.)

Hard disk drives

Two important numbers measure the performance of a hard disk drive. The access time, or seek time, is the time it takes the head of the disk drive to get to where the data is stored on the hard disk. This number is important because it affects the time it takes for things to start happening (such as for a movie to start playing). More importantly, if you are accessing a lot of information stored in different places, then a slow access time will affect performance. Most hard disk drives available today have access times in the range of 10 to 17 ms (milliseconds), which is the recommended range.

The second number is the sustainable transfer rate—the amount of information that can be transferred from the hard disk to the computer. If you are recording or playing QuickTime movies, then you will want a drive with the best performance possible. Unfortunately,
this number is not dependent just upon the disk drive. Many Macintosh models cannot support transfer rates higher than around 1,500 kilobytes per second because of the limits of the SCSI bus in those models. Some hard disk manufacturers have found a way of bypassing this problem by using a NuBus card that plugs into the Macintosh and transfers information through the NuBus hardware rather than through the SCSI hardware. As one example, Storage Dimensions sells a 1G (gigabyte) 2FAST SCSI disk drive that has a transfer rate for writing data of 2,097 kilobytes per second and a read rating of 2,419 kilobytes per second when used with a Macintosh IIci. This is very close to the rating of a Quadra.

**Removable media**

Several removable media options are available. The most common ones in use today are SyQuest, Bernoulli, and 128M optical disks.

SyQuest drives originally stored 44M on a cartridge, but newer models support 88M. The older drives cannot read the 88M cartridges. The first 88M drives could only read the 44M cartridges; they could not write to them. Newer drives can read and write to both types of cartridges. The cartridges themselves are about five inches square and three-fourths of an inch thick.

SyQuest drives have two advantages. Because of their popularity, both the drives and the disks are inexpensive compared with other removable media. The disks, however, are more fragile and more subject to errors than the other removable media mentioned here. If you need to send large amounts of material to service bureaus or customers, you might want to buy a SyQuest just because it is very likely that your customers will have a SyQuest drive.

Bernoulli drives and disks are made and sold by only one manufacturer, unlike SyQuest drives, which are made by one company but several different companies OEM these drives. Bernoulli disks are available in two sizes, which closely match those of SyQuest disks (both in physical size and storage capacity). These drives are not based on the Winchester technology used in SyQuest drives, but on a different design that is much more resistant to disk crashes and other media problems. If you are constantly moving your drive (even while it's running!), you should consider a Bernoulli drive. They are slightly more expensive than SyQuest drives, however, and not as widely used.

---

1 OEM: Original Equipment Manufacturer. For a curious reason, an OEM company simply puts their label on a unit made by another company.
The 128M optical disk drives have become popular over the past year, particularly as the price has come down to close to $1,000 for the drives. The disks are about the size of a 3 1/2-inch floppy disk, but double the thickness. The initial costs are higher than for a SyQuest or Bernoulli system, but the cost per megabyte is lower for these drives. Look for double-sized drives (256M) and even lower prices over the next few years.

Some other removable drive options are available, including Flopticals (which hold only 20M) and 600M opticals (which tend to be slow and expensive). Neither are as popular as the SyQuest and Bernoulli drives, and primarily for that reason they are not recommended.

**CD-ROM**

The CD-ROM medium stores large amounts of information (600M) on a distribution medium that is cheap to reproduce (costing less than $2.00 per disc when ordered in quantity). It has taken several years for CD-ROM to establish itself, but there now seems no doubt that CD-ROM can be used successfully for distributing commercial products. The cost of CD-ROM drives has come down, and Apple even includes a built-in CD-ROM in some models.

The slow transfer rate and access times of CD-ROM drives compared with hard disk drives does limit their application. New double-speed drives and disk caching software have improved performance considerably, and careful attention to design by the multimedia producer can mask some of these performance problems.

Creating a CD-ROM is surprisingly simple, particularly if you are creating a Macintosh HFS (Hierarchical File System—the system used for organizing and storing files on a Macintosh) format disc. All you have to do is place the files on a hard disk, arrange them how you want them to appear, and then send them to a CD-ROM pressing plant. The pressing plant creates a master (costing around $1,500 to $2,000) and then presses as many discs as you want for about $2.00 each. Prices vary from company to company and differ depending upon packaging and label printing options. All manufacturers create a check disc, which is a single copy of the CD-ROM that you can use for testing purposes. This disc costs around $300 and is well worth the money to make sure that the CD-ROM works the way you thought it would.

---

*Caching: The software saves extra information read from the disk into a piece of memory. This often saves time if the user wants to read something that has already been read, and seems to improve the performance of the drive, even though the speed of the mechanism itself has not changed.*
Creating a CD-ROM that is compatible with both Macs and PCs is also possible. Several commercial games and educational products are already available for both computers on a single CD-ROM. The CD-ROM pressing plant will provide you with specifications for preparing the disc and associated artwork. A brief list of manufacturers is included in the sidebar, “A short list of CD-ROM plants.”

A short list of CD-ROM plants

3M Optical Recording Department
3M Center Building 223-5N-01
St. Paul, MN 55144-1000
(612) 733-2142

DMI
1409 Foulk Road, Suite 202
Wilmington, DE 19803
(302) 433-2500

METATEC
7001 Discovery Boulevard
Dublin, OH 43017
(614) 761-2000

Nimbus Information Systems
Box 7305
Charlottesville, VA 22906
(804) 985-1100

Optical Media International
485 Alberto Way
Los Gatos, CA 95032
(408) 395-4332

SONY Electronic Publishing
1800 N. Fruitridge Avenue
Terre Haute, IN 47803
(812) 462-8160

CD-ROMs are usually pressed; however, systems are available that etch or burn information into special blank discs to create one-off CD-ROMs that can be used in regular CD-ROM drives. These devices
are expensive (currently around $4,000), and blank discs cost around $30 each. Some service bureaus create these discs for you (essentially they are offering the check disc service offered by the disc pressing plants but without the option to mass-produce the discs).

**Laser discs**

Although laser video discs might appear to be little more than oversized CD-ROMs, they are actually a completely different format for storing information. Laser video discs store only video and sound and provide that information not in a digital format, but in an analog format suitable for playback on a television monitor.

Laser discs have been used with computers to create interesting presentations. Professional laser disc players have a serial port that can be used to control the device. With the appropriate cable and software, the Macintosh can start, stop, and advance the player to a specific frame on the disc. Usually an external monitor displays the signal coming from the laser disc, which means that the user has to deal with two monitors—the Macintosh and the television. Figure 5.1 shows how a Macintosh, a laser disc player, and a television monitor work together.

**Figure 5.1.** When using a laser disc connected to the computer, the user tells the computer what to do and then watches the result on a television monitor.

It is possible to use a video display board for the Macintosh which can display video on the Macintosh screen. This configuration creates a seamless production, but it increases the cost of the equipment needed to play back the production because you need a Macintosh with a color monitor, and a special card that will display NTSC (the standard used for television transmission in the US) video on the Macintosh screen.
Using laser discs has both advantages and disadvantages. Access time from one end of the laser disc to the other can amount to several seconds, making it impossible to jump seamlessly from one part of the video to another. Integrating a laser disc with your production is much more complicated than working with hard disk drives or CD-ROM drives because of the extra software, hardware, and cables necessary to get the thing working.

The quality of the sound and video, however, is much higher than is currently possible with QuickTime (currently the only alternative if you need video in a multimedia production). Also, a laser disc can be used as a source for a large number of still images. Each frame of the video can be accessed independently, and an hour of video is made up of more than 100,000 still images. You can have a laser disc made for $300—there are a few service bureaus that do this, check in the back of MacWeek or MacWorld—which makes it quite an attractive medium for special installations.

The HyperCard VideoDisc Toolkit is a collection of XCMDs for controlling laser discs from within HyperCard. It is available from APDA for $40.³

**Still other hardware**


**The future**

Apple is already working on new Macintoshes that have built-in DSPs (digital signal processors), which will improve the sound and video digitizing capabilities of the Macintosh.
Apple is also working on versions of its System software that will run on other platforms. There is rumored to be a version of the System under development that will run on PCs. At the same time, Apple is working on the PowerPC—a RISC-based processor jointly developed by IBM, Motorola, and Apple. Apple is also moving in other directions with Kaleida, Taligent, and Newton.

What does this all mean? It’s too soon to tell where the Macintosh’s operating system will go—whether it will continue to evolve or become obsolete. Even less clear is what these newest projects mean for multimedia developers who currently work on the Macintosh. Still, it’s too early to start panicking. Apple is strongly committed to the Macintosh and will remain so for the foreseeable future.
You might wonder why a book about multimedia includes a chapter about text. Text is an important part of multimedia presentations and programs. Text is often the most concise and compact way of communicating a large amount of information—at times, it's the only way. Multimedia developers sometimes get carried away by the bells and whistles of sound and video, and forget that text might be conveying the most important information in the production. This chapter is here to remind you not to make that mistake.

The chapter begins with a brief history of text on the Macintosh—particularly how the Macintosh stores and displays text. If you already are comfortable with terms such as bitmapped font, point size, PostScript, TrueType, and ATM, you can skip the first section and go straight to the sections on what text says and how text looks. The following sections describe how hypertext (linked text in electronic documents) works, provide tips for using text in productions, and ends with an unusual use for text in your productions.

A brief history of text on the Macintosh

The first Macintosh was rather unique in the personal computer market in that it used bitmapped fonts to create text. When a computer uses bitmapped fonts, it stores each text character as a small graphic file. The character is defined by the arrangement of pixels onscreen. The computer stores a separate graphic for each point size (the height of a character) of each character in the font. The larger the point size of the font, the larger the size of the file that contains it.

To limit the amount of space required for storing bitmapped fonts, early Macintosh computers offered a limited number of font sizes—10, 12, 14, 18, and 24 points. Users could display text in other sizes, but the computer simply scaled one of its existing font sizes, so the result didn’t always look very good.

Early Macintosh computers also had a screen resolution of 72 pixels per inch (ppi) and stored the fonts at that resolution. (Most Macintosh monitors today have screen resolutions between 72 and 80 ppi.)
The fonts looked great onscreen, but didn't look so good when printed.

At that time the only printer that worked with the Macintosh was a dot-matrix printer that printed at 144 dots per inch (dpi)—a dot on a printer is like a pixel on the screen, so you can compare ppi to dpi. The problem with printing a 72 ppi font on a 144 dpi is that when the 72 ppi character gets to the printer it has half the information (or pixels) that the printer is capable of printing. The printer will print the character by scaling it. The character will look okay, but it won't look as smooth as the printer is capable of printing it.

To improve the resolution of printed type, the Macintosh sent characters of double the size being printed to the printer. If you were using 12-point type, for example, the computer sent 24-point (double the size) type to the printer. The printer scales the 24-point type to 12-point type, which creates better-looking printed characters (because the 24 point, 72 dpi character translates into a 12 point, 144 ppi character on the printer—see figure 6.1). This arrangement worked fine for text in 12-point type, but if you wanted to print 24- or 48-point type, then the text didn't look so good.

![Diagram of printing bitmapped text to a dot-matrix printer.](image)

Figure 6.1. Printing bitmapped text to a dot-matrix printer.
 Shortly after introducing the Macintosh, Apple entered into an arrangement with Adobe, a small company that had been developing a page description language called PostScript. Apple released a laser printer that printed at 300 dpi and used PostScript. Bitmapped fonts were of inadequate quality for use with this printer. Further, PostScript has its own font description that consists of a mathematical description of each character in a font. PostScript stores this information in a PostScript font file that can exist in ROM on the laser printer or on the Macintosh (in the latter case, the computer must send the font to the printer each time a document needs the font).

Figure 6.2 illustrates the process of printing a PostScript font with the Macintosh. In order to use a PostScript font in a Macintosh document, however, you must still install a bitmapped version of the font in the Macintosh System.

Figure 6.2. The relationship of bitmapped and PostScript fonts.

Note that figure 6.2 shows a laser (PostScript) font, called Helvetica, while the previous figure (6.1) shows a bit-mapped font called Geneva.

Although the first PostScript fonts looked great when printed, their onscreen display was limited to the bitmapped sizes because the Macintosh system still used the bit-mapped fonts when displaying text on the screen. To overcome this limitation, Adobe developed Adobe Type Manager (ATM). This product uses the PostScript font files to create any size font for display on the Macintosh screen. ATM is particularly helpful for viewing large fonts onscreen.

At about the same time that Adobe introduced ATM (perhaps there was a connection!), Apple announced a new type technology called
**TrueType.** In a sense, Apple designed TrueType to replace the bitmapped and PostScript files with a single file that can display a font at any size. TrueType entered the market as a competitor to the existing standard of PostScript fonts, and the publishing world was uncertain what effect the two products would have on each other’s sales and popularity.

At this writing, both products appear to hold their share of the marketplace. PostScript continues to be very popular in desktop publishing applications, and TrueType is distributed with many of Apple’s low-cost printers. The introduction of QuickDraw GX\(^2\) promises the addition of even more capabilities to TrueType. Many fonts are available in both TrueType and PostScript and, as a further benefit, you can use TrueType and PostScript fonts at the same time.

**TIP:**

The price of fonts has fallen dramatically in the past few years. Adobe sells a number of type sets that contain a selection of different display fonts at a very reasonable price. Adobe also sells Type On Call, a CD-ROM that contains all Adobe fonts. You must buy access to each PostScript font, but you get a bargain deal on one font. Best of all, the disc contains all of the bitmapped versions for all Adobe fonts, and you can use the bitmapped versions at no charge.

Several other companies offer collections of fonts as well, so most users can put together a reasonable collection for very little money.

---

**What text communicates**

Multimedia producers need to consider more than just the information text represents. They need to consider electronic features that can be added to text—like hypertext—as well as more down-to-earth considerations such as, “Will the user have the font I am using in my presentation?”
Text

Text might appear to be dull and boring. It doesn't have the excitement and appeal of sound and video, but it's important, and computers add new power to text that it never had before—the ability to link text together so that the user can quickly and easily find related information.

Hypertext

Hypertext is a term coined by Ted Nelson to describe the linking or threading of text to help communicate knowledge. In some respects, hypertext is like a computerized footnote. By simply clicking on a word, you can see more information about the word (see figure 6.3). But why limit the link to words and their description? Perhaps the link might be to every other occurrence of the word in the book, to other books, or perhaps to a bibliography. The possibilities are limitless.

That's why it's so frustrating that it's still amazingly difficult to create hypertext documents on the Macintosh. There is no program that makes it easy to do—about the only applications that do are HyperCard and SuperCard, and while you can create hypertext links in these programs, it's not as easy as it should be. HyperCard is not hypertext. (For more information on HyperCard, see Chapter 14, "Authoring environments.")

Figure 6.3. The basic idea of hypertext.
The HyperCard programmer can find what word in a text field the
user clicked on, but the programmer has to then use that word to
search for related topics. In other words, it’s up to the programmer
to create the links. HyperCard 2.0 has an added feature called
Groups, that enables the developer to group two or three words as
one term. When the user clicks on one of the words in the term,
HyperCard returns all the words in the Group. This makes it possible
to provide a link to the term “Ted Nelson,” without worrying about
whether the user clicks the word “Ted” or “Nelson.”

When you create a text-rich multimedia production, try to imitate in
your production the features of a book. Include, for example, a table
of contents, an index, and a bibliography. In addition to those fea-
tures, you might consider adding a dictionary or appendix with
definitions for selected terms. As a final step, you can add links
between words.

Some hypertext elements are logical design choices. Creating a table
of contents and an index as separate hypertext system elements that
users access through a tool palette or menu command is a logical—
and obvious—design decision. Other elements of designing a
hypertext system are less obvious. In designing the interface for a
hypertext system, determining how to enable the user to interact
with the features is the biggest, and most frequently experienced,
design problem.

You must deal with a number of issues when designing a hypertext
system. The following sections describe the two most important of
those issues.

**Getting around in hypertext**

All hypertext designers must determine how to implement links
within the hypertext system. An example of a simple hypertext
system link is as follows: If a user encounters a word about which he
or she wants more information, the user clicks on the word and the
information appears. If the link has only one destination—for ex-
ample, a dictionary definition—then a simple hypertext link can
display the information when the user clicks a word. But what if the
information is in more than one place? What if you need to create an
additional link in the text? If a word is referred to in only one other place in the text, then perhaps you can link to that place. But what if there are other references (see figure 6.4)?

The Xanadu system uses hypertext to link together a wide range of information sources.

If this book were running on a hypertext system, then you could just click on a word to get a definition or find other references for the word. Well, maybe in the next edition we can do that!

HyperCard doesn't really support hypertext, although it is possible to create hypertext-like systems by using the tools available in HyperCard. For example...

The term hypertext was coined by Ted Nelson to describe a way of linking different information together.

**Figure 6.4.** Hypertext poses the problem: Where to go from here?

You can choose either of two solutions to this problem. One solution is to have the user hold down a special key for each type of destination. For example, Command-clicking might go to the dictionary, and Option-clicking might go to a reference elsewhere in the text. The other solution is to list all possible links in a palette that appears when the user clicks on the word.
None of the linking solutions is completely satisfactory. Pointing and clicking is the easiest linking method, but inadequate for multiple destinations. Option- or Command-clicking takes the user to a specific place, but requires that the user knows which key to use to get to that place. The palette provides an easy way to go to any of multiple places and shows the user all the choices, but requires a lot of extra clicking if the vast majority of links are single links to the dictionary, (the user must click the word, then click the entry in the palette). The data in your hypertext system may help you decide how to implement linking features.

**Getting back to where you were in hypertext**

The next problem with hypertext is returning the user to his or her original position after the link is completed. If the user clicks on something and is taken to another part of the text, the user must have a way to get back to where he or she was. HyperCard provides two features for achieving this return. The most commonly used method is to place a button on the card which, when clicked, takes the user back to the previous card that the user was on. This button uses a command that is available from the Go menu in HyperCard to take the user back to the most recently viewed card. The other feature is the Recent window, which shows thumbnails of where in the text the user has recently been (see figure 6.5). The user can use the Recent window to jump back to a previous card.

![Figure 6.5. The Recent window in HyperCard.](image)
Both of these methods of returning to a link's starting point are helpful to the user, and you should provide access to them. The most important rule is to provide a clear way for the user to get back to where he or she came from in the text. If your hypertext system does not do this, the user wastes a lot of time making the return or, even worse, gets hopelessly lost in the effort.

### Using text in productions

The discussion of computer text and type uses some typesetting terminology in ways uncommon to other print media. The following definitions, therefore, may eliminate any confusion over the use of those terms in this chapter.

**Fonts, typefaces?**

In strict typesetting terminology a **typeface** is a specific style (for example, Helvetica Bold), and a **font** is a specific character size of a specific typeface (for example, 14-point Helvetica Bold). A **type family** is a collection of similar typefaces (for example, Helvetica, Helvetica Italic, and Helvetica Bold make up a type family). The Macintosh documentation, however, uses the term "font" to mean type family (for example, Helvetica is a font, and bold or italic are styles).

When using fonts in multimedia productions, you will often encounter the same questions that traditional graphic designers consider: Which font communicates the information? Which font looks best? In short, which font should I use?

Multimedia designers have an additional problem that they have to design for the computer screen. This means accommodating the smaller screen area (compared to a standard piece of paper) as well as the lower resolution of the screen—the lower screen resolution means that you usually have to use slightly larger fonts than would be possible on a printed page, since printers usually have much higher resolutions.
Fonts

The multimedia developer should have access to a wide range of fonts. As a designer, you must decide whether to use bitmapped, PostScript, or TrueType fonts. When selecting a font, remember that some fonts are common to almost every user’s System. Those fonts are: Geneva, New York, Chicago, Times, Helvetica, Palatino, and Zapf Chancery. You can use these fonts freely in your multimedia project designs without worrying that a user’s System doesn’t support the font you have selected. If you use almost any font not included in the previous list—whether it is bitmapped, PostScript, or TrueType—you face the potential problem that users might not have the font installed in their System.

If a user runs a project that includes a font he or she doesn’t have, the Macintosh substitutes another font (often Geneva). The user can see and read the type, but the substitution can often make the presentation difficult to read.

How can you avoid this problem? You could include the fonts you are using as a separate file with the presentation. The problem with that is that most fonts are protected by copyright. If you want to give a copy of them to someone you must license the copy.3

If you decide to include a copy of a bitmapped font, you must determine how to distribute the font with your production. Apple maintains that the best way to accomplish this distribution is to have the user install the font into his or her System file. This solution, however, is not always ideal, because it’s an extra step the user has to perform.

An alternative solution is to install the font into a document or application; when that document or application is open, the font is available. Apple does not recommend this procedure—primarily because of possible font conflicts. Apple also may fear that this method may not work in a future version of the System, and it causes problems if you try to print a document using a print spooler (when the spooler comes to print the document, the font may no longer be available.)

It’s up to you which method to use.

3The issue of font protection is complex because a typeface currently cannot be protected under copyright laws. Some of the program code used to generate PostScript fonts, however, has been granted protection. As a result of this protection, someone can create a font that looks like a protected PostScript font, but they can’t simply copy the PostScript file. Bitmapped font files are a different story. Although a company may want to claim a copyright on a bitmapped font, most companies (like Adobe) have allowed users to copy the bitmapped versions freely, so the users can create layouts at home and then go to a service bureau to print the final copy.
TIP:
If your production includes a large amount of text, I recommend that you distribute a copy of the font. This advice is particularly appropriate if you need only the smaller type sizes, in which case you can use a bitmapped copy. For larger headlines or display type, consider converting the text to a graphic. Although this technique consumes more memory (particularly if you are storing a lot of text this way), it does result in text that always looks right.

Some production designers may want (or need) to create their own fonts. You can create fonts in a couple of ways. If you need only a few bitmapped font characters, you can create them using ResEdit, Apple's resource editing program. ResEdit is available from user groups, online services, and APDA (Apple Programmers and Developers Association). ResEdit is not a simple program to use, and very little documentation exists for it. A good reference is the *ResEdit Reference*, published by Addison-Wesley. You can benefit from having a copy of ResEdit and knowing how to use it, because you can use the program to customize icons and cursors. Figure 6.6 shows creating a special character in a font being edited in ResEdit.

![Figure 6.6. Editing an iconic character in a font, using ResEdit.](image-url)
To create a PostScript or TrueType font, you must use a program such as Fontographer or FontStudio. These programs are designed for creating fonts, but also provide sophisticated drawing tools that rival those in Adobe Illustrator and Aldus FreeHand.

**Hyperfonts?**

You can use fonts to create unexpected things—such as buttons and icons. You can put just about any kind of graphic into a font, including signatures or other items that you might use many times.

A designer can create a control panel, for example, made up of a text field with a special font for the control panel’s buttons (figure 6.7 shows just such an example I created in a HyperCard stack). When the user clicks a “button” in the field, a script determines the character pressed and then reverses the character to provide a high-lighting effect.

![Figure 6.7. A font used to create buttons in a control panel.](image)

As the example in figure 6.7 above shows, there’s more to fonts that just text!

The next chapter discusses graphics. It has been said that a picture is worth a thousand words, so it is perhaps appropriate that the graphics chapter follows this chapter.
chapter 7

Graphics
Creating graphics on a Macintosh is a simple process. You grab the mouse and “draw” or “paint” with it just as though you were holding a pen or brush. As you drag the mouse, a “pen” or “paintbrush” moves onscreen and produces the graphic. Creating graphics on a Macintosh is so simple and obvious that it’s hard to believe this chapter is necessary.

Well, of course, creating graphics on a Macintosh isn’t that simple. It took a long time to get to where we are today, and there are a lot of choices when it comes to creating graphics on the Macintosh, including what format to save the graphic in, and (more importantly) which of the many programs to use to create the graphic.

A brief history of graphics on the Macintosh

In the beginning was the screen. (And if you look closely, you will notice that the screen is made up of individual “points” called pixels, which stand for “picture elements.”) It didn’t take long for someone to realize that you could create graphics by changing the color of the pixels. And so computer graphics were born. By the late 1950s and the 1960s, some reasonably sophisticated graphics programs were available for mainframe computers, but it took the arrival of the personal computer in the early 1980s to bring computer graphics into the mainstream.

Graphics programs can be split into two broad categories; bitmap-based and object-oriented (the latter are also sometimes called “drawing” programs just to confuse things). With bitmap-based programs, an illustration is created on a “page” that contains a predefined number of pixels (the resolution of the illustration). The graphic is created by changing the color of individual pixels. This provides a lot of flexibility, but is difficult to change elements once they have been added.
Object-oriented programs use mathematical equations to describe individual elements (such as a circle or rectangle). An illustration is created using these elements. It's easy to edit elements after they've been created, but it's more difficult to fine tune your illustration with an object-oriented program.

The first Macintosh graphics program was MacPaint. The first version of this program was designed for use with the early Macintosh computers, on whose screens pixels could be only black or white. MacPaint provided tools for drawing circles, rectangles, and lines, as well as paintbrush, airbrush, and eraser tools. Figure 7.1 shows MacPaint 2.0, the most recent version of this program.

Figure 7.1. MacPaint 2.0, the most recent release of the original Macintosh paint program. The pencil tool is being used to add details to the graphic.

MacPaint is easy to use, but very limited. The program creates graphics at 72 dots per inch (dpi) and a fixed size of 576 x 720 pixels, which produces a graphic slightly smaller than an 8 1/2- by 11-inch sheet of paper when printed at 72 dpi. The 72 dpi resolution matches the resolution of the Macintosh screen, so the graphic prints at the same size it appeared onscreen. This is fine when printing with the ImageWriter printer (the first printer for the Macintosh) because the resolution of the ImageWriter is only 144 dpi. But the low resolution becomes noticeable when MacPaint images are printed on a high resolution printer such as a laser printer (300 dpi).
MacPaint (and other bitmap-based graphics programs) have other problems. After you create a graphic, it is very hard to make changes. If you draw a line and then decide to move it just a bit, you have to erase the first line, pixel by pixel, and then draw a new line in the correct location. This is a very tedious procedure. Another problem is that MacPaint's graphics are limited to the resolution at which you create them—in fact, all bitmapped graphics are limited to the resolution at which you create them. If you create a graphic at 72 dpi, you cannot achieve better results by printing the graphic on a 300 dpi printer than you can by printing it on a 72 dpi printer. You're stuck with the 72 dpi resolution.

Enter MacDraw. This program uses objects (circles, lines, polygons) to create graphics. At first glance, this seems to be what MacPaint offered; but the difference is that when you draw a circle in MacDraw, the circle is saved as a coded description of the object. The object is defined as a circle of a fixed size in a fixed location rather than as a collection of pixels. You can easily move, delete, or change the size of the circle after you create it. Figure 7.2 shows how an ellipse can be manipulated after it is created.

![Figure 7.2. Resizing an ellipse in MacDraw.](image)

MacDraw supported a new file format, called PICT (for PICTure), in which you could save graphics that contained objects as well as bitmapped illustrations.
Creating a graphic with MacDraw is very different from creating a graphic with MacPaint. In MacDraw everything is based on a mathematical description—a line, a curve, or a polygon. This can constrain the effects that you can create. And you can't manually turn off or on pixels in the graphic. Also, the objects in a MacDraw graphic are stacked on top of one another. Although you can select an object and move it in front of or behind another object at any time, you constantly must deal with the relative locations of objects.

But MacDraw's object graphics are easier to edit than MacPaint's bitmapped graphics, and have other benefits as well. Because MacDraw saves the objects as coded information rather than as arrays of pixel values, MacDraw files are much smaller than MacPaint files. Even better, the resolution of the MacDraw objects depends only upon the device used to display or print the graphic. Onscreen, a MacDraw graphic looks as though it were drawn at 72 dpi. On a laser printer, the graphic is "drawn" at 300 dpi—the computer or printer simply adds more pixels based on the description of the object.

You might think that object-oriented graphics programs would have spelled the end of bitmap-based graphics programs. This was not the case (there is even a program that combines the features of a bitmap-based graphics program and an object-oriented graphics program—SuperPaint). First, no object-oriented program has yet been able to imitate traditional "painterly" effects as well as a bitmap-based painting program. Second, scanners create only bitmapped graphics, so if you use a scanner and need to manipulate the scanned graphics, then you will need a bitmap-based graphics program.

MacDraw and object-oriented graphics were a major step forward for some people, but others still struggle with pixels. Specifically, anyone who needs to save and edit scanned images. The original PICT format was not suitable for this task. (The original PICT format did not support the quantity of information that scanners produced. It has since been updated and now can be used to save scanned images.) To solve this problem, a group of software and hardware vendors got together and created an informal standard graphics format for large, multiple bit-depth images. This format, TIFF (Tagged Image File Format), is still widely used for scanned images.
Our history of Macintosh graphics continues...

Shortly after MacDraw appeared on the scene came the introduction of the Apple LaserWriter. The LaserWriter holds an important place in history: by enabling the desktop publishing revolution to occur, the LaserWriter literally saved the Macintosh from oblivion. The LaserWriter also introduced PostScript to the Macintosh graphic artist.

PostScript is a page description language. It is the language the computer uses to tell the printer what to print. The Macintosh uses its own language to draw onscreen—QuickDraw. The printers used with the Macintosh before the LaserWriter were thus QuickDraw printers, and the Macintosh used QuickDraw to create the pattern of pixels it sent to the printer. QuickDraw was good for screen printing and low resolution printing, but wasn’t as sophisticated as PostScript. One important difference is the basic drawing element of each language. QuickDraw uses polygons to create complex shapes. PostScript uses Bézier curves (a mathematical description of a curve based on two anchor points and two “handles,” which you move to change the shape of the curve). The Bézier curves not only are the basis of the fine resolution possible with the LaserWriter; they also became the basis of a new generation of drawing programs—the PostScript drawing programs. (See Chapter 6, “Text,” which also discusses PostScript and QuickDraw as they relate to text.)

Not long after the success of the LaserWriter, Adobe, the creators of PostScript, released a program called Illustrator. Illustrator was both similar to and very different from MacDraw. Like MacDraw, Illustrator was an object-oriented program. Anything that was created in Illustrator was treated as an object that retained its identity and could be edited at any time. But whereas MacDraw was based on QuickDraw and used polygons to create shapes, Illustrator used Bézier curves (see figure 7.3).

Included on the disk is a demo version of Illustrator.

Although Illustrator is more complicated to use than MacDraw, it is much easier to create very sophisticated graphics with Illustrator; it has become (along with the competing product, Aldus FreeHand) the standard graphics program for graphic designers.
Figure 7.3. Adjusting a Bézier curve in Illustrator.

One particular advantage Illustrator and FreeHand have over MacDraw (and similar programs based on QuickDraw) is that they are more accurate when printing to PostScript printers. PostScript did not exist when QuickDraw was designed, so QuickDraw code has to be converted to PostScript code in order for a QuickDraw graphic to print on a PostScript printer. This conversion often causes slight problems (for example, lines that don’t quite align the way you expect them to align). These problems don’t occur with PostScript drawing programs. (Of course, if you are creating a graphic for display onscreen, then this is not a problem.)

Both Illustrator and FreeHand save the graphics they create in **EPS** (Encapsulated PostScript file) format. A PostScript file is simply a file containing the PostScript code needed to print a page or pages. This file can be sent directly to the printer using a utility, and the printer prints a page containing the graphic. Alternatively, the graphic can be imported into a page layout program and then printed as part of the layout. The difference between a PostScript file and an EPS is that the EPS contains a low resolution bitmapped version of the PostScript file (that is, what the graphic looks like). This feature is used in page layout programs to provide previews for positioning the graphic, but they are comparatively low resolution representations of the graphics.
PostScript files are principally useful for printing on PostScript printers. They are virtually useless for printing to non-PostScript printers. Many people like to use Illustrator or FreeHand to create graphics, however, and there's no reason why you shouldn't use these programs to create an illustration that will be seen only onscreen.

In the last few years, a new file format has become popular for saving bitmapped graphics. JPEG (Joint Photographic Experts Group) is a file format specifically designed for compressing photographic images. Unlike some compression formats that reduce the size of a file but always can recreate the original file exactly, JPEG is "lossy." A lossy compression format throws away some information; however, if you use the format correctly, you will not notice that anything has been removed. JPEG provides a sliding scale of compression, making it possible to increase the compression factor (which reduces the quality of the image and the size of the file) or reduce the compression factor (and increase the quality of the image and the file size.)

JPEG has its detractors. Some people get upset at the very idea of throwing away any information. But I find it impossible to tell the difference between the original and JPEG-compressed files, at least when using the minimum compression settings (and even at the minimum settings, the space savings are considerable). Describing the JPEG compression process in depth is beyond the scope of this book, but there are other references available for those who are interested (including the QuickTime Handbook, published by Hayden).

Apple included a JPEG compressor in QuickTime. If your computer can run QuickTime, then you can use QuickTime to compress images in the JPEG format. This is particularly useful if you have to send files over a modem.

**Some tools**

Today MacPaint has been replaced by several programs that are designed for manipulating scanned images or provide very "painterly" effects. Object-oriented graphics programs have split into two camps. On one side are programs such as MacDraw Pro and Canvas,
which provide the features found in the original MacDraw and some additions. On the other side are the PostScript graphics programs Adobe Illustrator and Aldus FreeHand.

Which object-oriented graphics program should you get? If you are doing screen graphics or simple printed graphics, then you can choose any or all of the applications available, since the final output is at comparatively low resolution. If you are a serious desktop publisher or graphic designer who is typesetting your output, then you should consider Illustrator or FreeHand, though you will probably end up with several other applications as well.

The following sections summarize the types of graphics applications available, as well as the different file formats that they use.

**Graphics editing programs**

For manipulating bitmapped graphics (such as scanned images), the most popular program is Adobe Photoshop (see figure 7.4). ColorStudio offers many similar features. Unfortunately, ColorStudio was marketed initially by Letraset and did not receive as much attention as Photoshop. ColorStudio now is published by the creators of the program, but has lost considerable ground to the competition.

Figure 7.4. Adobe Photoshop.
If you want to create "painterly" illustrations, then you should look at Fractal Designs Painter (see figure 7.5). This program does a very good job of imitating traditional painting tools such as chalk, charcoal, and different paintbrushes. It's an amazing program, particularly if you have a pressure-sensitive drawing tablet (discussed later in this chapter).

Figure 7.5. Fractal Designs Painter.

Included on the disk are demo versions of Photoshop and Painter.

Basic object-oriented graphics programs

For object-oriented graphics, two of the most popular programs are Canvas from Deneba and MacDraw Pro from Claris. There’s also SuperPaint from Aldus, which supports both a bitmapped and an object layer. This feature provides a "best of both worlds" combination, except that the individual functions are not as sophisticated as the dedicated applications. If you are working only occasionally in either "medium," then SuperPaint might be a good choice.
PostScript graphics programs

The two most popular PostScript drawing programs are Adobe Illustrator and Aldus FreeHand. Both offer a similar selection of tools and functions, and it often comes down to a personal preferences choice. I might prefer the way the pen tool "feels" in Illustrator, and you might prefer the way layers containing objects can be built up in FreeHand.

Graphs or graphics?

There are some other programs that you might use to create graphics which aren't really graphics programs at all. In particular, spreadsheet programs like Microsoft Excel provide charting functions for creating very sophisticated charts. The latest version creates all kinds of 2D and 3D charts. The only problem I find with Excel is that you have to be pretty sophisticated yourself to figure out how to use it! However, a program like Excel can be very useful if you need to create charts and graphs, because even if the chart it produces isn't exactly right, you can use the basic charts Excel creates as the basis to create and manipulate the charts in another program.

File formats and applications that work with them

There are a lot of graphic file formats out there. Fortunately, though, most programs support several formats. You shouldn't find exchanging graphics with another person too difficult—even if that person has a graphics program that is different from yours.

MacPaint

The MacPaint format is still widely used, although no program today supports only the MacPaint format. Many bitmap-editing
applications, such as Photoshop and Studio/32, read MacPaint format, as do all desktop publishing programs (though none of the latter group will let you edit MacPaint graphics).

**PICT**

PICT is widely used for bitmapped and object graphics. It no longer has the limitations in color, resolution, and size that the original format had.

**TIFF**

The TIFF format was originally created for scanned images. The most common programs for manipulating TIFF images are Photoshop and ColorStudio.

---

**Tip:**

Should you use the TIFF format? Although TIFF is still widely used, it has some disadvantages. The greatest disadvantage is that TIFF does not compress files in any way, resulting in files much larger than those saved in PICT format. (A compression option that considerably reduces the size of a file has been added to the TIFF standard, but the option is rather slow and increases the time it takes to open and close the files.)

If you are working in desktop publishing, then you might want to use TIFF. In the past, I have found that some desktop publishing programs (such as PageMaker) handled the TIFF format much better than they handled PICT; however, this is not so much the case today. If you aren’t involved in desktop publishing, then it’s probably better to use PICT. Most of the advantages of TIFF have disappeared as a result of improvements in the PICT format.
RIFF

This is a compressed bitmapped format first used by the grayscale editing program ImageStudio; RIFF is also used in ColorStudio. The RIFF format is not widely used, although several programs support it.

EPS

The EPS format is supported by Illustrator and FreeHand. You also can save bitmapped graphics in EPS format (but see the tip earlier in this chapter).

Remember that EPS is designed for sending files to a printer, not for exchanging files for editing. It is generally not possible to use Illustrator to edit drawings created in FreeHand and vice versa.

**Tip: When to use EPS**

Although the above discussion may imply that EPS files are only for object-oriented graphics, you can save bitmapped graphics in EPS format. Saving bitmapped graphics in EPS format creates large files that are difficult to work with, however, so you should use the EPS format for bitmapped graphics only when you have a specific purpose in mind—for example, for printing on a PostScript printer (and even then, using the EPS format is not always necessary). Saving a bitmapped illustration in EPS format will *not* enable you to edit the file in Illustrator or FreeHand.

Targa

The Targa bitmapped format comes from the PC platform, and though several programs support this format (including Photoshop), Targa is really just for those moving files from the Mac to the PC and back again.
GIF

The GIF (Graphics Interchange Format) file format is a compressed file format for bitmapped images. CompuServe developed it for distributing graphics over modems. It has become fairly popular, and several programs support this format although it does not support bit depths greater than 8 bits.

Included on the disk is the program GIFConverter (see figure 7.6). You can use this utility to convert GIF format graphics to a wide range of other formats, as well as to convert those formats into GIF files.

Figure 7.6. GIFConverter.

JPEG

The JPEG format was developed by an independent group, and although it is a "standard," this does not mean that one program that reads JPEG files will be able to read the JPEG files created by another program (although developers are slowly eliminating these inconsistencies).
When JPEG compresses photographic images, it throws away some information. This loss, however, is usually not noticeable. Because JPEG enables you to adjust the compression factor, you can control the loss to some extent. If you compress the file to the maximum amount, you reduce the quality of the image more, but if you reduce the compression factor, you retain more of the quality of the image.

The JPEG compressor that Apple includes with QuickTime is available to anyone who can run the QuickTime extension. Included in the QuickTime Starter Kit (a collection of utilities sold with QuickTime by Apple) is a utility called Picture Compressor, which can be used to compress images.

Included on the disk is a copy of QuickTime 1.6

As shown in figure 7.7, the first step is to open an image in Picture Compressor. The next step, shown in figure 7.8, is to choose the compressor (Photo-JPEG) and the compression quality. Selecting Compress from the Compression menu starts compressing the image, and a new window, shown in figure 7.9, displays the resulting image.

Figure 7.7. Compressing using Picture Compressor.
As you can see in this example, a 1M PICT file was reduced to 100K by using the Normal compression quality. Choosing Most would have increased the savings even more (though at the expense of image quality). After an image is compressed, it is saved on a disk. Other applications recognize the new file as a PICT file, so any existing
application that can open a PICT file can open this compressed file. (When the application asks the System to open the file, the System recognizes that the file is compressed and decompresses the file before providing the data to the application.)

Existing applications can read QuickTime-compressed images (though they will not be able to create them), but you must have QuickTime to read these files. Also, the files are not in a strict JPEG format, so if you want to send the file to another computer system, additional translation work is required.

Included on the disk is a copy of the shareware program JPEGView (see figure 7.10). This program can open and display JPEG images.

![JPEGView](image)

Figure 7.10. Viewing a JPEG image in JPEGView.

**Other sources of graphic images**

So far, this chapter has focused on applications for creating and editing bitmapped and object-oriented graphics. There are, however, other ways of creating graphic images for your multimedia projects.
Video images

If you have (or are considering obtaining) a video camera for recording source material for QuickTime movies, you might consider using that camera as a source of still images. All that is required is a video digitizer—and you probably have (or are considering obtaining) a digitizer for capturing the video anyway.

Be aware that some video digitizers are better than others. For example, the VideoSpigot from SuperMac is great for capturing small images (up to 320 x 240 pixels at 72 dpi resolution), but cannot be used for digitizing larger frames unless the source image is still. This is because capturing a large image takes several frames to complete. The digitizers from RasterOps (the 24STV, 24 MXTV, and so on), as well as Radius’ VideoVision, are all capable of digitizing a large single frame. RasterOps includes a plug-in extension for Photoshop that can be used to capture images and import them straight into Photoshop for editing (see figure 7.11).

Figure 7.11. Grabbing an image with a RasterOps digitizer using a Photoshop plug-in.

If your digitizer can’t digitize a single frame, you still might get good results if your video source has a clean pause (that is, the video image remains stable when the source machine is placed in pause mode).
Although a video camera can be a quick and convenient tool for grabbing still images, it has certain drawbacks. For example, nearly all cameras require a second or so to start and stop recording, which makes quick “grab” shots almost impossible. When digitizing an image from videotape, you need several seconds of the subject to cue yourself to click the “Grab” button. For this reason, you might want to use other sources for digitizing a large number of images.

Still video images

Still video cameras have been available for several years. They use a CCD (charged couple device) to capture an image, much the same way camcorders capture a frame of video. The images are stored on a small magnetic disk, however, rather than on video tape.

Two versions of the cameras are widely available. The “professional” units cost several thousand dollars and store 25 images on a single 2.5-inch video floppy disk. The less expensive “consumer” units, which cost around $500, store 50 images on a disk. The consumer cameras store more images by halving the vertical resolution of the images. This increases the number of pictures per disk, but diminishes the quality of the images.

After an image has been captured, it can be displayed on a television, or digitized by using a video digitizing card (just like digitizing images from a video camera or tape). Unfortunately, the consumer cameras, although attractively priced, produce very poor quality images. This poor quality is primarily because of the low resolution of the images, but other factors such as the encoding system used to save the images on the disk also affect the quality. Generally, the consumer cameras should be used only for images of 320 x 240 pixels or smaller.

Although the professional cameras produce acceptable results at 640 x 480 pixels, the images are not as sharp as those produced by scanning photos or slides. This difference is most noticeable if the images captured with a still video camera are used alongside images from other sources like a slide or flatbed scanner.
Some professional still video cameras do not store their images on the 2.5-inch floppy disks. Rather than convert the image to the analog signal used on the 2.5-inch disks, these cameras store the data digitally, which improves the quality of the image. These cameras also offer improved resolution thanks to higher-resolution CCDs. Most notable among these are Kodak's DCC-200 series cameras. These cameras are based on the Nikon 8008S camera body, and include a special back that contains the CCD and a hard disk for storing images. After images are captured, the camera is attached to a computer via a SCSI cable, and the images are downloaded to the hard disk.

The DCC-200 series cameras produce high quality results, but they are extremely expensive (around $10,000). If you are taking more than five images per day, the camera will pay for itself, but if you do not generate that number of images, you might be better served by other methods.

### Scanned images

Flatbed scanners and slide scanners are the most common choices for digitizing still images. The prices of both pieces of equipment
have come down considerably in the past five years. Devices that used to cost around $10,000 are now available for less than $2,000.

It is possible to get a single pass, 24-bit color flatbed scanner for less than $1,500. This is amazing. In selecting one of these scanners, the decision often comes down to features and software. Several scanners include a version of Photoshop with a plug-in for Photoshop so that you can drive the scanner from within the Photoshop application. This feature makes scanning and manipulating the images very convenient. Other scanners, such as the Hewlett-Packard Scanjet llc, have their own applications. A good third-party application for driving scanners is Ofoto from Light Source, Inc. (see figure 7.13). This program not only drives the scanner, but also provides several controls for adjusting images and functions for calibrating the scanner. If you scan a lot of color images, you may want to consider Ofoto.

![Figure 7.13. Adjusting the brightness of an image scanned into Ofoto.](image)

If you are considering buying a slide or flat bed scanner, you should first look at Kodak's Photo CD system (discussed in the following section). The Photo CD system may be more cost-effective than a scanner if you do only a few scans per week and probably will provide images of much higher resolution than a scanner (although the resolution is not as critical for most multimedia developers, because they are generally working only with screen resolution images of 640 x 480 pixels).
Kodak Photo CD images

Last year Kodak unveiled the Photo CD system, with the primary intention of selling it in the consumer market. Photo CD also has received a lot of attention from desktop publishers, graphic artists, and multimedia producers.

What is Photo CD?

Back when still video cameras first appeared (see above), Kodak began to worry that this type of camera would spell the death of film. So, the company set about creating a system that provided the features of a still video camera (your pictures on TV) while preserving film.

How does Photo CD work?

You take a roll of film and expose it in the traditional way—using your camera. Then you take the exposed film to a photo processor and ask the processor to develop the film and provide a Photo CD. The photo lab develops the film, then digitizes the images and stores them on a CD, much like an audio CD or CD-ROM (except that the information is in a different format). Up to 100 high-resolution images can be stored on one Photo CD. The photo processor also can print traditional photographs from the film negatives.

When you come back to pick up your photos, you receive the Photo CD and film negatives. You take the Photo CD home, insert it into a special CD player (which also plays regular audio CDs) and watch the images on your television. It's simply marvelous, and it's simply hard to believe that Kodak thought a lot of people would want to look at their photographs on television.

Still, the Photo CD contains all these images digitized at 3072 x 2048 pixels, and Kodak has released programs for reading Photo CDs on CD-ROM drives attached to computers. Here is where Photo CD has found its biggest market. For between $1 and $3 per image (depending upon where you go), you can obtain images scanned at high resolution and stored on an easy-to-handle medium—a CD.
Apple has even licensed the Kodak decompression routine and added it to QuickTime 1.5. When you insert a Photo CD into a computer running QuickTime 1.5 and the Apple Photo Extension, the images on the disk appear as thumbnails (small graphic representations of the images); they can be opened and viewed using any application that reads PICT files. Figure 7.14 shows a Photo CD displayed in the Finder. The bottom left window is a QuickTime movie that shows all the images on the disk.

![Figure 7.14. A Kodak Photo CD.](image)

**How do single-session and multi-session drives differ?**

If you already own a CD-ROM and want to use Photo CDs with it, then you need to know about single-session versus multi-session discs. The Photo CD format supports multiple sessions. You can add one roll of film to a Photo CD, then come back two weeks later and add a second roll of film to the same Photo CD, and continue this until the disc is full. When the first roll of film is put on the disc, that Photo CD is a single-session disc. If you add another roll of film at a later date, then that Photo CD becomes a multi-session disc.
Nearly all CD-ROM drives sold before 1992, and some still being sold, are only “single-session compatible.” If you insert a multi-session Photo CD into the CD-ROM drive, the drive will read the images from the first session, but it won’t read the additional images.

This limitation is not a major problem as long as you don’t create a multi-session Photo CD when you have only a single-session-compatible CD-ROM drive. That’s why it is important to check your drive. Of Apple’s CD-ROM drives, only the 300 series drives support multi-session disks. The 150, SC, and SC Plus drives do not support multi-session Photo CDs. To find out about other manufacturers, refer to the instruction manual or contact the manufacturer.

Clipped images

For those who don’t have the time or talent to create their own graphics, a wide selection of clip art of different types is available. Refer to Chapter 12, “Clip media (and copyright),” for more information on this source of graphics.

Pressure-sensitive drawing tablets

Pressure-sensitive drawing tablets don’t really fit in this section, but I had to squeeze them in somewhere! If you have a talent for drawing—or even if you just like doodling with a paint program—then you should definitely consider buying a pressure-sensitive drawing tablet. Several programs, including Fractal Designs Painter, support these tablets. When you draw with a pressure-sensitive tablet, you can adjust the width or the flow rate or some other characteristic of the line you are drawing by varying the pressure with which you draw. The pressure-sensitive tablet makes it possible to imitate traditional drawing media much more closely than is possible using just a mouse.
I highly recommend using a pressure-sensitive tablet. Be warned, though, that it helps to have a fast computer (a 128i is barely fast enough) to work with these tablets and the paint programs that support them. The computer must perform extra processing to create the complex lines that you draw.

The future

In the future it will become easier to sample and manipulate the world in which we live.

As computers become more powerful, video and still images will become harder to distinguish. A need for still images will remain, but motion will tend to swamp other forms of media. Digital effects will become more sophisticated, but designers will continue to search for that new look. Nothing gets easier; it just changes.

Now that you have mastered the world of single images, the next chapter explores a world filled with hundreds and thousands of them—animation.
Chapter 8
Animation
Until the arrival of QuickTime and digital video (see Chapter 10, "QuickTime"), simple animation programs were the only method for creating motion on the Macintosh. For many purposes, animation programs remain the best way to create motion. From animated diagrams to cartoon movies, animation tools play an important part of the multimedia producer's arsenal.

Today's Macintosh users can choose between any one of several animation tools, but this was not always the case. For the first few years of the existence of the Macintosh, the only available animation program was a product called VideoWorks, which eventually became Macromind Director (see figure 8.1). Now there are several different programs available, from the low-end animation packages such as PROMotion and Animation Works, to the animated presentation programs Cinemation and Motion Works. Director remains the preeminent animation package on the Macintosh—and is described in much greater depth in Chapter 14, "Authoring"—but it now has some stiff competition.

Figure 8.1. Creating animation in Director.

This chapter is devoted to an overview of the techniques and problems you might encounter when working with animation; it also provides an overview of the many animation packages available for today's Macintosh user.
Working with animation

Working with animation on the Macintosh (or on any computer for that matter) involves becoming familiar with the techniques and problems of animation. When you understand these techniques you will be better prepared to solve the problems presented by the animation process. The information in this section outlines some animation techniques and some of the problems you must solve in order to make full use of your animation application. This section also discusses formats in which you can store animation files.

A few principles of animation

Two divergent schools of thought dominate computer-based animation. Members of the first school are those who use the available 3D modeling and animation tools to create 3D animation (see chapter 9, "3D," for more information on 3D modeling). Most animators who use those programs strive to create the most realistic animation possible. To achieve that realism, the motion of animated objects must closely model the movements of real objects.

The second school—the traditional animation school—is rooted in the long history of cartoon animation. It is this second group of animators who will be most interested in the programs discussed in this chapter. As the sophistication of the 3D programs increases, these two schools will diverge even more, because the ultimate goal of 3D modeling is to recreate real life using a computer. If you are interested in 3D animation you should skip to the 3D chapter.

Several principles govern cartoon animation, most of which center around the techniques of exaggeration. Movement, actions, and reactions in cartoons are always exaggerated, because cartoons themselves are exaggerations.

Anticipation is a specific application of exaggeration. Before moving in one direction, a cartoon character draws back slightly in the opposite direction. This backward movement makes the character look as though he is gathering himself together before he starts off, and results in a much more realistic movement than if the character simply started forward.
Squash and Stretch are used to show the effects of acceleration on an object. When a ball hits the ground it squashes down; it then stretches out as it rebounds. Objects do this in real life, though we rarely notice it. Figure 8.2 shows a party hat demonstrating the effects of Squash and Stretch.

Figure 8.2. An object squashing and stretching.

A number of books are available for those interested in learning classic animation. One book I like is *The Animators Workbook*, by Tony White. This book's tutorial format leads you through animation principles and shows several examples. There are also a number of books on Disney and Warner Brothers cartoons which would serve as good introductions to cartoon animation.

As a multimedia producer, you might use animation programs frequently, and yet never use any of the classic animation techniques. Often, animation in multimedia presentations involves very simple actions such as animating a cursor in a program demonstration, or animating bullet points onto the screen during a presentation.

However, even in those cases you might learn things from the classic animation styles. For example, rather than have a bullet point simply slide off the screen, you could use the anticipation technique of moving the item fractionally in the opposite direction before the object starts sliding off the screen.

You should also study the motion of objects in the real world to try and recreate fluid motion in your animations. Many animation programs let you determine the starting and ending points of an object; the animation program will then calculate the in-between
steps to create the movement of the object across the screen. There are two problems with this. First, the movement is in a straight line from point A to point B. Second, the object will instantly begin moving and will instantly stop. There will be no apparent acceleration of the object. This makes the movement of the object appear very unrealistic.

Instead, you should try to vary the path and speed of the object. Some animation programs let you do this (Director provides a special command for doing this) but you have to use these functions. Usually the default is to create the simple movement.

Computer-based animation can result in beautiful—and quite complex—productions. As with any computer tool, the more capabilities a program has, the more you need to learn in order to use the program productively. The following section discusses some problems unique to computer-based animation and offers some solutions to those problems.

A few problems of animation

Synchronizing sound with animation is one of the most difficult tasks facing any animator. Even if you create animation in the traditional method and record the results to film, you must spend much time and effort in order to synchronize the sound with the animation.

You might think that this task is easier on the computer—but it isn't. Unfortunately, the computer often exacerbates the problem rather than solving it. The problem with sound synchronization on the computer is that sometimes the computer is unable to play the frames of the animation at the correct speed to maintain synchronization with the sounds. This is particularly true when animations created on faster machines are played on slower machines. You can also create an animation that is too complex for your hardware to play back at the frame rate you desire. For example, I can create an animation in Director that I want to play at 30 frames per second, but if the animation involves a lot of objects moving around the screen, then the animation might only play at 10 or 15 frames per second.
Once I have created the animation at 30 frames per second, if I try to synch a sound track to the animation I will have a problem, because the animation will take twice as long to play as it should. I then either have to increase the length of the sound track or remove frames from the animation, in order to synchronize the animation to the sound track.

You need not record the sound track as one long piece. Instead, you can cue individual sound bites to specific events in the animation. This approach reduces some of the synching problems, but not all of them; it's not feasible, for example, to cut what a character speaks into individual words and then cue the words to the opening and closing of the character's mouth!

A second problem in computer-based animation is memory management. Before most animation programs play an animation, they load all the elements used in creating the animation into memory. With all the elements loaded in advance, the program has immediate access to all the animation's graphics and can quickly display them as they occur during playback of the animation. The problem with this arrangement is that your animation must fit within your computer's available memory; further, if you link two animation files together, the program will spend several seconds loading the second animation before it begins playing.

Apple's QuickTime software has effectively eliminated the animator's synching and memory problems. QuickTime (described in Chapter 10, "QuickTime") provides a mechanism for maintaining synchronization between animation and sound, no matter what computer is playing the animation. QuickTime doesn't suffer from memory problems because it can read from disk while playing animation on the screen.

QuickTime has its limitations, however. Currently, QuickTime does not support interactivity (while several animation programs do). Further, because QuickTime saves each frame individually, its files can be much larger than animation program files (which save only individual animation elements). And finally, slower computers may not be able to play back QuickTime animations that fill the whole screen.
File formats

Prior to QuickTime's release, only two formats for moving animation information from one program to another were in common use: a folder of PICT images and the PICS format. When saving an animation as a folder of PICTs, each frame of the animation is saved with the same name plus a unique sequence number at the end of the filename (for example, TEST 0001, TEST 0002, TEST 0003, and so on).

The PICS format was an attempt by several developers to manage the problems inherent in shifting animation around as separate images. Apple didn't develop the PICS format, but PICS became a de facto standard for the Macintosh because it provided an easy method of linking several PICT files into a single file. Nearly all programs that create animation can save the animation in PICS format.

The PICS format includes an optional method for reducing the size of the files. This option enables the program that creates the PICS file to save only the changes that occur from one frame to the next (see figure 8.3). This compression can considerably reduce the size of the file. Any program that reads PICS files should be capable of recreating the original frames.

![Frame 1](Image)

![Frame 2](Image)

Animation

PICS

(difference between frames 1 and 2)

Figure 8.3. Reducing the size of PICS files by saving only the frame difference.
QuickTime may soon replace the PICS format as the standard for exchanging animation information. Many programs, however, include both PICS support and QuickTime support. This approach makes sense, because many users still can’t run QuickTime.

The QuickTime Animation compressor is provided specifically for compressing animations. The compressor uses run-length encoding to compress animation. Run-length encoding essentially looks at an image and stores the number of duplicate pixels as a count (for example, 20 pixels with the same color begin in row 5 at pixel 33), rather than as separate values. This form of compression works well with animation, which is frequently made up of large areas of the same color.

Some animation tools

In the last couple of years, several new applications for creating animation have appeared on the market. These applications offer different tools and methods for creating animation, and several are a lot cheaper than Director. Although Director still may be the best choice for some people who want to create animation on the Macintosh, it is no longer the only choice. This section provides a glimpse at some of the animation applications currently available for Macintosh users.

Life Forms

Macromedia’s Life Forms is the most unusual of the tools listed here because it is designed for animating one thing only—the human body. Life Forms provides a wireframe model of the human body (see figure 8.4). You animate this form on the screen to create a sequence of movements. To create the sequence, you set the form in one position, advance the animation frame counter several frames forward, and then adjust the positions of the form by clicking and dragging. You don’t have to set the position in every frame—Life Forms creates the in-between frames to complete the motion. This technique is called key frame animation (Chapter 9, “3D,” contains further information about key frame animation).
Because Life Forms animates the human body only, the parameters of movement for each part of the body are built into the model—you can’t move the body into impossible positions. Although some might consider these built-in parameters a limitation, the parameters make creating the animation sequences much easier!

When you complete a Life Forms animation, you can export it to a 3D program for rendering. Programs supported by Life Forms include Swivel 3D, Electric Image, and MacroMind 3D. You also can save the animation as a PICS file for importing into other animation programs. If you use this option, you can add other elements or redraw Life Forms’ simple outline output. For example, the juggler shown in figure 8.5 was created in Life Forms, the globes were added in Director, and the completed animation was then exported to QuickTime. Life Forms also includes an XCMD for playing back Life Forms animations within HyperCard, Director, or any other program that supports XCMDs.

The QuickTime movie Juggler (described in the text above) was created using Life Forms and is on the CD-ROM.
PROmotion and ADDmotion

Developed by Motion Works, PROmotion and ADDmotion are essentially the same program—PROmotion is a stand-alone application, and ADDmotion is a set of XCMDs that run within HyperCard. The XCMDs enable you to create and play animations within HyperCard stacks, thereby adding the animation capabilities of ADDmotion to the scripting capabilities of HyperCard. The programs differ slightly—PROmotion can save animations as stand-alone documents and has a print to video feature which blanks the rest of the screen while an animation is playing (useful when recording to video tape.) Both programs, however, provide the same tools for creating animation.

TIP:
Beginning in the summer of 1993, PROmotion will be marketed and distributed by Corel Corporation, under the name CorelMOVE.

Both PROmotion and ADDmotion use paths and actors to create animation. You use either program's built-in paint program to create an actor (a graphic that is to be animated), or you import the actor from PICT or PICS files. An actor can consist of several different cels...
The eels may show the actor in a series of different positions—for example, running. Actors are animated over backgrounds—still images which are imported into the program and placed on the screen. You place the actor on the screen, and then animate the actor by drawing a line or path on the screen using special tools. You can alter this path at any time. During playback, the actor cycles through its eels while it follows the path. Figure 8.6 shows manipulation of the path an actor moves over in PROmotion.

Figure 8.6. PROmotion.

A demo version of PROmotion is on the CD-ROM.

Because ADDmotion (the original program) is designed to work with HyperCard, its animations can be interactive. The animation can stop at different points and pause for a specified amount of time, or until the user clicks an actor. Clicking also can cue events. In ADDmotion, this cue sends a call-back to HyperCard (in other words, when you click on an actor, you issue a message to HyperCard that tells it to do something). PROmotion supports AppleEvents—a new capability added to the Macintosh system, which allows one program to send messages to other applications, telling them to perform specific functions. You can control a wide range of functions from a PROmotion animation when you use it with a program such as UserLand's Frontier.
You can export animations you create in PROMotion to other applications. PROMotion includes support for QuickTime movies, an After Dark format file, and a self-playing movie format. You can distribute these stand-alone animations free of charge.

The PROMotion and ADDmotion animation applications are inexpensive and provide many features. The biggest problem with both programs is that creating animations with them is rather slow—saving and opening animations and actors takes several seconds—on all but the fastest Macs.

**Animation Works**

Animation Works, from Gold Disk Software, is very similar to PROMotion and ADDmotion. The animation in Animation Works is made up of actors that are made up of cels. You animate the actors along paths on the screen to create the final animation (see figure 8.7). Animation Works includes some XCMDs for playing animation in HyperCard.

![Figure 8.7. Editing path-based animation in Animation Works.](image-url)
Although Animation Works, ADDmotion, and PROmotion are similar programs, each has benefits over the others for specific uses. If you are interested in using animation with HyperCard, then you may prefer ADDmotion because animations can be used to control HyperCard functions. If you want to control AppleEvents, then PROmotion might be your choice because of the built-in AppleEvent support. If you want to create path-based animation only, Animation Works might be the best choice because its performance is better when editing and manipulating animations.

Cinemation

Cinemation is an interactive animation environment that includes several tools that help you create animations quickly and easily. You can drag an object around the screen, and Cinemation records the movement and then plays it back. You also can specify key frames and have Cinemation create the in-between frames. Cinemation also can use a loop of an animation, such as a figure walking, to create an animation sequence.

Cinemation has two primary windows. You draw and manipulate objects in the Movie window, which also displays the animation. The Filmstrip window shows small thumbnails that represent the different frames of the animation; you can click on a thumbnail to advance to the frame it represents. Figure 8.8 shows the Cinemation screen; in the figure, a globe is in the Movie Window and a series of thumbnails of the globe is in the Filmstrip Window.

A demo version of Cinemation is on the CD-ROM.

Cinemation was designed to animate presentations. You can import a PowerPoint or Persuasion presentation into Cinemation and then add animation effects, such as sliding bullets, to the presentation. You can export Cinemation movies to QuickTime or PICS files.
MovieWorks

MovieWorks is also designed for creating animated, interactive presentations. MovieWorks is actually four programs combined into one animation package. You use the Text, Sound, and Paint Editors to edit the separate media types. You use the Composer application to combine these elements to create QuickTime movies or interactive presentations. Composer enables you to integrate the text, sound, and paint materials, and to specify timing relationships between the presentation’s elements.

To create a presentation you must import the presentation elements into MovieWork’s Medibase Window and then arrange them on the stage. The tools provided in the Tool Palette are used to create paths along which the different elements of the animation move. Each element of a movie is contained in a separate track. You can view the tracks in the TimeView window, which also shows where you have applied effects. Figure 8.9 shows the manipulation of a path used to animate an object in MovieWorks.

A demo version of MovieWorks is on the CD-ROM.
Other animation packages

Some other animation products are worth knowing about. Animation Stand, from Linker Systems, is an animation tool that operates on the traditional cel-based principles of animation. In Animation Stand, you record the position of every object in every frame on a spreadsheet (traditional animation calls this the **dope sheet**). This program is expensive, and is really designed for the professional who outputs animation to film or video tape.

The FilmMaker animation program was developed by a French company and sold by ParaComp. Since ParaComp and MacroMind merged to form Macromedia, however, little advertising or news about FilmMaker has been released. FilmMaker is path-based, though the interface is much more complicated than the other path-based programs mentioned above. FilmMaker is a sophisticated product, and I hope Macromedia will either start promoting the product or sell it to another company.
Another program that disappeared when ParaComp and MacroMind merged is Magic. This program originally sold as an interactive presentation tool that offered animation functions. When Macromedia released Action! (its new interactive presentation tool) the company repositioned Magic as an animation tool; not much has been heard about Magic since then.

Finally, don’t forget Director, which remains a very popular and powerful package that can be used to create interactive animations. You should read Chapter 14, “Authoring environments,” for more information about Director.

The future

The future of animation on the Macintosh will be exciting—a safe prediction now that QuickTime provides a very good medium for distributing animations. Interest in, and the demand for, 3D tools that create animation is growing—probably because of all of us who can’t draw well enough to create animation in the traditional way! The following chapter discusses some 3D modeling and animation tools available for use on the Macintosh.
It took a long time for the Macintosh to find a niche in the 3D world. Until the Macintosh II arrived, 3D and the Macintosh warranted little consideration—the hardware wasn't powerful enough and the screens were only black and white. You couldn't use the Macintosh to create photorealistic images. Now Macintosh users suffer from an embarrassment of riches—more than a dozen different applications perform some kind of 3D modeling, rendering, or animation.

If you work in multimedia then you need to know at least the principles of 3D programs, if only so you know what's involved in creating a 3D logo for your presentation. Once you decide you need to create 3D images, you'll discover that a vast number of 3D programs are available, and you should have little trouble finding a product that suits your needs.

This chapter begins with a discussion of the basic functions performed by 3D applications and the features these programs offer. The following section discusses how to choose between the many programs available, and the chapter finishes with an overview of several popular packages.

The three functions of 3D modeling

3D applications perform three functions: modeling, rendering, and animation. **Modeling** is the process of creating the different objects that make up a scene. **Rendering** is the process of reproducing the scene in a photo-realistic manner. **Animation** is the process of creating a sequence of frames that show the objects moving within a scene.

When choosing a program, it's important to realize that some programs perform all three functions, and others perform only one or two. For example, Ray Dream Designer enables you to create models and render a single image, but has no facility for creating animation. Pixar's ShowPlace is an animation and rendering package that provides no modeling tools—you must import models that are created
in other programs. Pixar’s MacRenderMan is a rendering program that does nothing but render images from scenes created in other programs. Specular International’s Infini-D and Strata’s StrataVision provide all three modeling program functions.

Should you buy a program that provides all three 3D functions (and save some money) or should you buy two or three specialty programs? The answer to this question depends upon what you want to do with 3D, as well as the functions provided by the different programs. For example, although Infini-D and StrataVision are good all-purpose tools, you may prefer MacroMind Three-D’s animation process. Alternatively, you might not want to create animation, and find that Ray Dream is all you need. The modeling tools offered in Sketch! and MacroModel may be more to your liking than those offered in the other programs. The section “Considerations when choosing 3D applications” will help you in making these decisions.

The following sections detail the three functions of 3D applications in more detail.

**Basic modeling**

All 3D modeling programs contain tools with which you can create some basic objects, such as blocks, spheres, pyramids, cones. You can use these objects to create simple scenes only; you cannot use them to create complex scenes. Figure 9.1 illustrates the basic object tools available in StrataVision.

![Figure 9.1. The basic object tools in StrataVision.](image)
Extrusion

The simplest 3D modeling method is extrusion. When you create an **extrusion**, you draw a two-dimensional shape and then extrude or push out the shape in the third dimension to create a 3D shape (see figure 9.2).

The complexity of the extruded shape depends upon the tools the program provides for drawing the 2D shape. Some programs provide only a polygon drawing tool; others can draw smooth curves. Several 3D programs enable you to import a PICT file and use it as a template for the 2D shape.

![Figure 9.2. Extruding a 2D shape in Ray Dream Designer.](image)

Extrusion is particularly useful for creating logos. Several programs, such as Pixar’s Typestrix, enable you to add bevels to extruded objects to further enhance their 3D appearance (see figure 9.3).
Lathing

You create a lathed object by sweeping a 2D shape around an axis of rotation to create a 3D shape (see figure 9.4). Lathing results in much more complicated objects than those you can create with extrusion. Lathing is particularly useful for creating objects such as bottles, cups, light bulbs, or any other shape that is symmetrical around an axis.
Sweeping

Sweeping (sometimes called "lofting") is a hybrid of the lathing tool. Sweeping enables you to scale the 2D shape as well as move the shape up or down the axis of rotation as you create the 3D object. The nautilus shell shown in figure 9.5 is an example of an object created with sweeping.

Figure 9.5. Sweeping a shape in StrataVision 3D.

Freeform

Although you can use extrusion and lathing to create a wide variety of shapes, some shapes are impossible to create using those tools. Nearly all 3D programs provide some kind of freeform modeling that enables you to manipulate an object in three dimensions.

The concept of freeform modeling can be difficult to grasp. The processes of extrusion and lathing are reasonably easy to visualize, but the concept of working in three dimensions can be confusing—particularly since the computer screen has only two!

The "traditional" method for freeform modeling is provided in programs like Infini-D and Swivel 3D. These programs provide three
planes (views) of the object—usually top, side, and front—and enable you to manipulate the object in each of the planes. The results of your manipulations are shown in an isometric (three dimensional) view. Infini-D’s freeform modeling screen is shown in figure 9.6.

![Top view of object](image1)

![3D (isometric) view of object](image2)

![Front view of object](image3)

![Side view of object](image4)

**Figure 9.6.** Creating a free form object in Infini-D.

**Advanced Freeform**

Even for the most practiced user, the modeling tools discussed so far have limitations. You can use them to manipulate only certain edges of the object—you can’t just grab part of the object and pull or push it to make the shape you want. To overcome that limitation, you need a tool that simulates clay modeling.

Enter Sketch! Sketch!, from Alias Research, uses a different method of creating shapes. Each shape is made up of many control points; you can grab, drag, or move any one of these control points to change the shape. Figure 9.7 shows a shape being created with the Sketch! 3D modeling program.
Figure 9.7. Manipulating a shape in Sketch!

Another program which offers similar functionality is Macromedia's ModelShop. Although Sketch! and ModelShop enable you to create very complex shapes, its functionality comes at a price. They both require a lot of memory and a fast processor.

Exchanging Models—DXF format

Even if one program doesn't create the shapes you need, you may be able to create the shapes in another program and then import them. While all 3D modeling programs use their own proprietary format for saving 3D information, one standard format does exist for interchanging 3D information—DXF. This format was originally designed for transporting files between CAD applications. DXF supports both 2D and 3D scene descriptions, but the general nature of the format means that various companies have implemented the format in slightly differently ways. As a result of these slight differences, just because two programs both say they can save and read DXF format does not mean that you can move models quickly and easily from one program to the other. Further, because DXF files tend to be much larger than models saved in a program's native format, you may create a model that is so large and complex that the importing program cannot read and import it. To avoid this problem, test to determine whether or not you can transfer models between programs before you begin working on a project where this is a requirement.
Rendering

3D modeling programs with rendering capability use mathematical algorithms to create lifelike 3D objects. Most of these programs provide some kind of high quality photo-realistic rendering algorithm, and nearly all support several different algorithms. Some are fast but low quality, while others are slow but produce much better results. These algorithms split into two camps—shading algorithms take into account the surface color of an object and other variables to work out what color should be applied (shaded) on an object. Ray Tracing algorithms trace the path of light in a scene, creating color on object as the path of the light rays is calculated.

Images created with one program may have subtle visual differences from images created with another program—even when both programs offer similar rendering quality.

One criticism leveled by some potential users of 3D applications is that the renderings look “too perfect” and too obviously computer-drawn. Even the most sophisticated programs suffer from this problem; it may be some time before computers can produce images that are indistinguishable from photographs.

When deciding which rendering capability you want in a 3D modeling program, you may want to consider how much time the program requires to render an image. Infini-D, for example, can create stunning images with Ray Tracing but takes a long time to do so. You could, of course, also use Phong Shading in Infini-D, but that results in lower-quality images. Luckily, you can speed the rendering process in some programs—including Infini-D—by using either network rendering engines or plug-in accelerators (these devices are discussed later in the “Network and accelerated rendering” section of this chapter).

Shading

The shading algorithm displays the rendered image; the algorithm used determines the time required to render the image, as well as the image’s quality.
Wireframe

When you edit and manipulate models, most programs display the objects either as a **bounding box** (a rectangle representing the size of the object) or a **wireframe** (an outline of the object's basic shape). Figure 9.8 illustrates an Infini-D wireframe display. Using these display styles speeds redrawing and improves the program's performance while you work with the model.

![Wireframe](image)

**Figure 9.8.** Wireframe representation of a scene created in Infini-D.

Flat shading

With **flat shading** the program renders each facet (or polygon) that makes up the surface of the object with a single flat color. This shading technique results in sudden color changes from surface to surface and accentuates the individual surfaces that make up the object (see figure 9.9). Flat shading is very fast, and you can use it to give a quick indication of the colors in a scene.

![Flat shading](image)

**Figure 9.9.** Flat shading representation of a scene created in Infini-D.
Gouraud shading

Gouraud shading is a technique in which the 3D rendering program calculates the colors of each surface at the surface's vertices and then interpolates the colors across the surface. This shading technique produces a smearing of color across each surface to connect subtly with the color of the adjoining surface (see figure 9.10). Because the surface colors do not change abruptly, Gouraud shaded objects look much more realistic than flat shaded objects.

![Gouraud shading representation of a scene created in Infini-D.](image)

Phong shading

Phong shading is a smooth shading algorithm developed by Phong BuiTuong. Phong shading takes into account three characteristics of a surface: diffusion, specularity, and shininess. Using information about these characteristics and about the scene's lighting, the algorithm calculates the color value for each pixel of the surface. Phong shading can create very smooth surfaces that have reflectivity, shadows, and transparency (see figure 9.11).

Note that a scene that would take a minute or so to render in Gourand shading would take several minutes with Phong shading.

![Phong shading representation of a scene created in Infini-D.](image)
Ray Tracing

Ray tracing calculates the path of light rays from their sources in the image to the viewer's eye. Ray tracing figures in reflection and transparency, and creates extremely realistic scenes (see figure 9.12). Rendering a scene with ray tracing is a slow process. A scene that is rendered in 10 minutes with Phong shading may take 100 minutes or more to render with ray tracing.

![Ray Trace Image](image)

Figure 9.12. Ray-traced representation of a scene created in Infini-D.

Ray Trace Image, included on the CD-ROM that comes with this book, is the image reproduced in figure 9.12.

Surfaces

While the rendering algorithm is largely responsible for determining the quality of a final image, another determining factor is the software's surface modeling capabilities.

Procedural surfaces

Nearly all 3D programs offer some kind of procedural shader. A procedural shader is a mathematical description for a surface that defines such properties as the shininess and transparency of the surface. The advantage of procedural shaders is that they can accurately illustrate an object that is made from a solid material—for example, a sphere carved from a piece of wood appears to have grain that runs realistically through the sphere.
The problem with using procedural shaders is that, in most cases, you cannot write your own shaders and are thus limited to those that come with the package. Luckily, most shaders have parameters that enable you to adjust the qualities of the surface—for example the frequency and color of the grain in the wood to create a range of different surfaces. You can write your own shaders for MacRenderMan, but you must first learn the RenderMan render language (which resembles C code).

**Texture Mapping**

Texture mapping enables you to import a flat two-dimensional image (for example a PICT file) and wrap it around an object to create a surface. With this rendering technique, you can add just about any surface to an object simply by scanning a picture or drawing a surface in a paint program. Programs usually enable you to control the orientation of the surface as you wrap it around the object. This capability is particularly helpful when you want to apply a specific pattern to a surface (like a logo on the side of a spaceship). Figure 9.13 shows texture mapping being applied in the Ray Dream Designer program.

![Texture Mapping in Ray Dream Designer](image)

**Figure 9.13.** Applying a texture map in Ray Dream Designer.

You also can use texture maps to create bump maps, reflection maps, and transparency maps. A **bump map** creates the illusion of raised
areas on an object’s surface. You can use a bump map, for example, to depict continents on a globe, as illustrated in figure 9.15. The figure shows two renderings of the same object (a globe). Both renderings use the same texture map, but a bump map is applied to the object on the right. Both renderings in this figure were done in MacroMind Three-D.

![Figure 9.14](image1.png)

**Figure 9.14.** Specifying the texture map and bump map in MacroMind Three-D for figure 9.15.

![Figure 9.15](image2.png)

**Figure 9.15.** Two renderings of an object using the same texture map, but with a bump map applied to the rendering on the right.

The raised areas shown on the globe’s surface don’t actually exist on the model, but are effects added by the bump map. Most bumps maps are merely gray scale images. The whiter the pixel in the image, the higher the “bump” on the surface (you can reverse this effect in some programs).

Reflection and transparency maps work in a similar fashion. The reflection map alters the reflectivity of the surface of the object (one part could be highly reflective, while another part would be less reflective); transparency maps perform the same function for the transparency of an object.
Lighting

Lighting plays a major role in the realism of scenes—a fact often overlooked by those starting out in 3D. Creating realistic lighting is not a trivial exercise, as anyone who has tried to light a real life scene for photography or videography can tell you. You often must add several lights to a scene to simulate daylight.

Perhaps the biggest problem in learning how to light a scene is that you must render a scene at the highest quality to see the lighting's complete effect on your scene. This can take a long period of time, and makes it difficult to experiment. To overcome this problem, beginners should experiment with lights in a very simple scene that can be rendered quickly so that they can easily see the effect of lighting changes.

3D programs usually offer three types of lights: **point lights** cast an even light in all directions; **spotlights** cast light in one direction only; and **ambient** light provides an overall level of light in the scene. You can light your scene with ambient light, but because ambient light usually does not cast shadows, you will have to add extra lights to make your scene look natural. Figure 9.16 shows the three types of lights as they affect a scene.

![Figure 9.16](image)

Figure 9.16. The same scene rendered with different lighting. Top left has ambient lighting only. Top right has point lighting with ambient lighting, and the bottom image has Spot Lighting with ambient lighting.
Positioning spotlights in 3D scenes can be tricky. Some programs provide a "point at" feature that enables you to point an object at a scene. This feature can be useful for positioning spotlights (see figure 9.17).

Figure 9.17. Adjusting a light in StrataVision. As you drag the light around the screen, a line from the light indicates the direction the light is pointing.

**RIB format and RenderMan**

As mentioned earlier in the chapter, some 3D modeling programs do not include a rendering function. When you create objects and scenes with such programs, you use other programs—called renderers—to render the scenes. You must save the scenes in a format that can be used by the renderer.

A de facto standard format for saving scenes to be rendered by a renderer is the RenderMan RIB format (a rendering package developed by Pixar). RenderMan has its own language for defining a scene as well as for writing procedural shaders. A modeling application saves a RIB file for rendering by a RenderMan-compatible renderer. Pixar sells MacRenderMan, a Macintosh version of the RenderMan renderer, although they also sell rendering packages for several other platforms.

The quality of the RenderMan renderer has helped make RIB a standard rendering format of the 3D world. Several programs save
files in RIB format, but many others include their own high quality rendering engine and do not support RIB. A lack of RIB support does not mean a program can’t be useful in its own right.

**Network and accelerated rendering**

The time required to render a scene has deterred many users from undertaking large tasks using Macintosh 3D packages. Several developers have recognized this fact and have come to the rescue with network rendering solutions. Network renderers send out parts of a scene’s data file to several computers on a network. Each computer renders a portion of the image and sends the results back to the master rendering program. The master rendering program puts the parts together to create the final image. Most network renderers break an image into small squares; as a rendering engine finishes a square, the network renderer sends it another, until all the squares in the image are complete. Network rendering greatly speeds the process of rendering a scene.

Network rendering engines are available for Ray Dream Designer, Infini-D, and StrataVision. The engines must be purchased separately for each machine on the network. Figure 9.18 shows network rendering in DreamNet, the Ray Dream Designers network renderer.

![Network rendering using DreamNet.](image)
Another way to speed up rendering times is to use a hardware accelerator. The YARC (the reverse spelling of "CRA Y") company makes a plug-in accelerator board that works with some software products to accelerate rendering times. Infini-D and MacRenderMan both work with the YARC board. Plug-in boards like the YARC board use a special processor and require that the developer rewrite the software to work with the board. Another way to decrease rendering times is with an accelerator card, an example of which is the Radius Rocket. These products contain a faster version of the Macintosh chip. Such boards speed up all Macintosh functions, not just rendering times. However, a YARC board can be three or more times faster than a single Rocket at rendering a scene.

Animation

Several programs provide animation features, though some make it much easier to create and manipulate animation than others. The following sections discuss some animation features and the methods used to create animation.

Creating animations

There are two primary methods for defining animation in 3D programs. The simplest method is called key frame animation. In key frame animation, you choose a particular frame (say, frame 10) in an animation, and instruct the program to record the position of all the objects in the frame. You then choose another frame (say frame 25), adjust the position of the objects and have the program record the position of the objects in the new frame. The program animates the scene by calculating the in-between positions of the objects from one key frame to the next.

The advantage of working with key frame animation is that it is easy to create, and the program can easily calculate and generate it. The disadvantage is that you can have difficulty editing or making subtle changes to the animation. Simulating acceleration of objects over time is particularly difficult in key frame animation.
Several programs now offer event-driven animation. With event-driven animation, you choose any object at any frame in an animation and move the object to a new location. This creates a new event that is specific to that object. Adjusting this event for this object does not alter the events or movement of other objects. Event-driven acceleration also makes it possible to add complex motions and change them easily. Figure 9.19 shows event-driven animation being created in the Infini-D program.

![Event-driven animation in Infini-D](image)

**Figure 9.19.** Event-driven animation in Infini-D.

**PICT, PICS, and QuickTime**

3D animation programs export their results in a number of formats. The most popular formats are a series of PICT files, a single PICS file, or a QuickTime movie. The PICT files format enables you to edit the images in a graphics program, while QuickTime is perhaps the best
way to distribute and play animations. With the advent of QuickTime, the PICS format has become less widely used, though many programs still support it.

Now that you are familiar with the features and functions available in 3D applications, you are better able to determine which program will be useful for your work. Each 3D modeling application is a unique blend of the functions and capabilities described in this section. The following section provides more information to help you choose an application that fits your needs.

The categories of 3D applications

To make the task of choosing a 3D application easier, it helps to first divide the many 3D applications into three broad categories: architectural modeling 3D applications; simple 3D extrusion applications; and general purpose 3D applications. By categorizing applications in this way you can at least eliminate some programs as you try to decide which program to buy.

Note that not every program fits into this categorization—MacRenderMan does not fit because it is a special purpose rendering program. However, if we created enough categories to completely enclose every product, it would not make sense to divide the programs into categories at all! Also, as a rule, 3D CAD programs have not been popular with Macintosh users, so they are not included in this discussion.

Architectural modeling 3D applications

Architectural modeling applications are specially designed to simplify the task of creating models of buildings and cityscapes. Two such applications, Macromedia's ModelShop and Dynaware's DynaPerspective, provide precise dimensioning of the objects that
makeup a 3D model. Both programs also provide functions useful for architectural models, such as the ability to create shadow studies (display where the light falls when the sun is in a particular location). Although neither application has very sophisticated rendering support, you can export models to another program to render the final scene. Both applications can create animated fly-thrus—an animated movie where the camera appears to fly through the building—for export to QuickTime or an animation program. Figure 9.20 shows the DynaPerspective program in use.

Figure 9.20. Designing a table with a plant in DynaPerspective. Elements like this are used as building blocks to create final scenes.

A hybrid program from Virtus, Virtus Walkthrough has most of the same characteristics of the 3D architectural modeling applications. Walkthrough enables you to create models in a way similar to ModelShop and DynaPerspective, but it offers one spectacular advantage to those programs. With ModelShop and DynaPerspective, rendering a fly-thru animation takes a long time; the programs can't produce a fly-thru in real time. Virtus Walkthrough can render the scene fast enough that you can change the view of the scene and immediately see how things change. The program can provide the illusion of walking through a scene (see figure 9.21). The quality of the Virtus Walkthrough rendering does not rival that offered by the other 3D modeling applications, but the realtime feedback provides other benefits.
Figure 9.21. Virtu Walkthrough. The window on the left shows the model being explored, while the window on the right is the rendered view of the scene.

You can use Virtu Walkthrough to create stand-alone versions of models for distribution to others. Other users can view and explore the models, but cannot change them. Virtu is considering adding other functions to the Player version. If they add buttons that play QuickTime movies or initiate other actions, then Walkthrough would be particularly useful to multimedia developers.

**Simple extrusion 3D programs**

Simple extrusion programs make up a relatively new category of 3D applications. These applications provide only the extrusion method of modeling (see the section entitled "Extrusion" that appears earlier in this chapter). You draw a two-dimensional object, then extrude it into the third dimension to create a three-dimensional object. Most of these programs are primarily designed for working with type, but some enable you to extrude other shapes as well.

Extrusion modeling is easy to understand and work with (the programs are simple for developers to write, as well). And since many people use 3D applications for the sole purpose of creating 3D logos, these programs are just right for that purpose.
Even though applications in this category provide extrusion options only, the quality of their renderings can equal those of general purpose 3D applications. Pixar's Typestry program includes a built-in version of the MacRenderMan rendering engine (see the discussion of this Pixar rendering software earlier in the chapter). MacRenderMan produces extremely high quality results. Typestry also can create a simple flying logo—3D letters and company logos that fly across the screen—and export the results to a QuickTime movie. Figure 9.22 shows a scene being created in Typestry.

Figure 9.22. Creating a scene in Typestry.

**General purpose 3D applications**

In terms of categorization, the general purpose 3D applications are all the programs that don't fit into the other two categories. Not to confuse the issue further, you could say that all 3D applications are general purpose tools, and that the other two categories of 3D applications are merely subsets of this category. Examples of general purpose 3D applications are Infini-D, Ray Dream Designer,
StrataVision, and MacroMind Three-D. These programs—and many others like them—have functions beyond simple extrusion and are not designed specifically for architectural modeling and can therefore be used as general purpose 3D modeling tools.

Considerations when choosing a 3D application

Several 3D applications currently share the market, and each has its own strengths and weaknesses. Choosing the right application can be the difference between owning a productive software tool and owning a box of software that sits untouched on the shelf.

Even within each category of 3D applications you are faced with a number of choices. Although the programs vary in the ways they enable you to create models, how they are manipulated, and in the quality of final renderings, the fact remains that you can create the same scene in many of the programs and achieve similar results. Choosing a program may come down to personal preference, the kind of application you have in mind, or some particular feature that you need.

For beginning 3D users, choosing a program can be particularly difficult. Take the time to consider the uses to which you may put the program. Do you intend to use animation? Do you plan to animate logos only? Are you looking for a neat effect, or a serious 3D modeling tool? If you can answer those questions, you may be able to narrow the field, from a dozen or so possibilities to a few strong contenders.

Other good sources of information are trade shows and colleagues. Approximately once a year one of the Macintosh magazines has a comparison of several 3D applications. Be wary of computer stores, where the salesman probably won’t be familiar with all the features of each product that is available.
The following section provides details about some of the 3D modeling applications currently on the market. The information in these sections can help you further understand the capabilities of 3D modeling, as well as the individual strengths and weaknesses of the listed programs.

A look at some 3D applications

To help those starting out in the world of 3D, the following sections briefly describe some of the programs currently available and highlight some of the advantages of each program.

Ray Dream Designer

Designer is a good all-purpose tool for creating still images. It does not support animation, but you can purchase network rendering software for speeding up the process of rendering. Designer does not support Ray Tracing.

Designer is a two-application set consisting of Light Forge, in which you create objects, and SceneBuilder, in which you arrange the objects into a scene that is then rendered.

Although Light Forge doesn't have the freeform editing capabilities offered by other programs, because of the way Light Forge displays objects as you create them, the application is much simpler to use than most other three-dimensional modeling programs.

Figure 9.23 shows an object being extruded in Light Forge. In the figure, you see the 2D shape drawn in the left window, and an extrusion line drawn in the upper-right window. The result of the extrusion is shown as an isometric 3D shape in the bottom-right window. You can click and drag on the 3D view to change the orientation of the shape, to enable you to view the results of your modeling from different angles.
You can create more complex objects by defining a series of different shapes along the length of the extrusion; Light Forge then creates a "skin" over those shapes (see figure 9.24). You step through the shapes in the left window using the two button arrows in that window's tool bar.
You define the object’s surfaces in Light Forge, then save the object and import it into the SceneBuilder application.

In SceneBuilder, you arrange the objects you created in Light Forge into a scene (see figure 9.25). The objects appear in an isometric view. Although moving objects in a single view can be difficult (considering the mouse moves in two directions only), Designer uses projections of the images on the three walls around the scene to help you control on which axis the movement occurs—you simply click on the projection of the object on one of the walls to move it along two axes. For example, clicking on the projection of the object on the back wall enables you to move the object up and down and side to side, while clicking on the projection on the floor lets you move the object forward and backward and side to side.

Figure 9.25. Arranging a scene in SceneBuilder.

Ray Dream Designer is a reasonably powerful program, and its low price (around $300 by mail order) makes it a very good value. Although Ray Dream doesn’t provide freeform object editing, the other tools make up for this omission. The program’s rendering quality cannot match that of a program such as Infini-D, but is certainly acceptable for most illustration purposes.

If you are new to 3D work, Ray Dream’s simplicity of use and low cost makes the program worth your consideration.
Specular International Infini-D

Infini-D from Specular International is a good all-purpose 3D application with general purpose modeling tools, a high-quality rendering engine, and powerful animation features. If you want to buy only one program, then Infini-D might be the one for you.

Perhaps the biggest problem with Infini-D is its lengthy rendering time, but Specular has addressed this problem by producing a network rendering engine (called BackBurner) that also supports the YARC co-processor board. You can create beveled text in Infini-D, and the latest release has special "filters" you use when creating animation effects.

Infini-D includes several procedural shaders and access to all of the shader parameters through a standard dialog. You can use PICT images as a surface map, or even use a QuickTime movie mapped to a surface. When you map a QuickTime movie to a surface, Infini-D uses the successive frames from the movie as a surface on the object in each successive frame of the animation. Figure 9.26 shows the Infini-D interface.

Figure 9.26. Infini-D. The top left window shows the side view of the scene, while the top right window is the camera view. The bottom right window is the Side view of the scene, while the bottom left window is the animation sequencer—used to animate the scene.
Strata Inc’s StrataVision 3D

StrataVision 3D’s modeling, rendering, and animation features rival those of Infini-D. Although Infini-D probably has the edge in animation, some users prefer the quality of the rendering produced in StrataVision. The two programs interface very differently, however, and you may prefer one program’s interface over that of the other.

Other differences between Strata and Infini-D are that Strata offers a sweep tool and a variation of the Ray Tracing algorithm, called Raydiosity. Raydiosity takes into account inter-reflections among objects and is especially suited to scenes with indirect lighting. Raydiosity renders even more slowly than Ray Tracing, however.

Strata sells a large number of texture maps as well as simple model libraries. Figure 9.27 shows a scene being manipulated in StrataVision.

![Figure 9.27. Manipulating a scene in StrataVision.](image)

MacroMind Three-D

MacroMind Three-D is a rendering and animation tool. The primary strength of this program is its animation capabilities. MacroMind
Three-D provides tools for creating simple shapes such as spheres and cubes, but you cannot use the program to create complex shapes.

Although MacroMind Three-D uses a score similar to those used in Infini-D and StrataVision for creating animation, the information provided in the MacroMind Three-D score is much more detailed. In MacroMind Three-D’s score, you can view numerical values for the orientation and location of an object and alter those numbers.

MacroMind Three-D is actually three programs. You create the scene in one program and use a separate rendering application to render the scene. The third application works like a post-production studio, in which you can matte animations with other images to create a finished scene. Matting is the process of merging an image of an object that does not contain a background with another image to create a complete scene. This capability is particularly useful for building with complex scenes, because you can render only the parts that change and then merge them together (which saves rendering time). Figure 9.28 shows an animation being created in MacroMind Three-D.

Figure 9.28. Creating an animation in MacroMind Three-D.
On the down side, MacroMind Three-D requires a powerful computer to provide good response time when manipulating scenes. It can take several seconds to respond to a user's action. If you can afford a Quadra, however, or are willing to put up with the lag in response time, then you might prefer the interface used for creating animation in MacroMind Three-D.

**Alias Sketch!**

Alias Sketch! is a modeling and rendering engine that lacks animation features. Sketch! more than compensates for this limitation, however, with its powerful tools for creating three-dimensional shapes. The capability to draw a freeform shape, extrude the shape, and then simply grab a point and pull it to deform the shape became available to Macintosh 3D users only with the release of Sketch!. Figure 9.29 shows a figure being modeled in Sketch!.

![Figure 9.29. Modeling in Alias Sketch!](image)

Sketch! also has a great feature for matting a 3D object into an image—for example, a photograph. You can open a PICT file, and then manipulate the object so that it fits within the scene; Sketch! provides
tools for defining to which plane the object is oriented. After you position and light the object, Sketch! renders the object to create a final scene that contains the background picture and the added object.

Be aware that you must have lots of memory and a fast processor to use Sketch!.

**All the rest?**

No user can expect to learn—let alone buy—every 3D modeling program that is currently available for the Macintosh. The applications mentioned above are those that the author has the most experience with; you may find that one of the many other available products better suits your needs.

Some other programs to consider include MacroModel (which provides freeform modeling similar to that in Sketch!), Electric Image (a popular application for television and film production due to its rendering engine speed), and ShowPlace (an easy-to-use program for arranging objects to render a scene using Pixar's MacRenderMan).

**Hardware requirements**

3D modeling applications are some of the most demanding applications available both in memory and processor requirements. You need at least 8M of memory, and some programs require nearly 20M when using complicated models. Even with these amounts of memory, rendering and manipulating complex models takes a long time on even the fastest Macintosh.

Most of these programs require an FPU (Floating Point Unit). The FPU is not available on all Macintosh models, but may be an option. If you think you may someday want to work with 3D, make sure you buy a computer that has an FPU.

You may need a hardware accelerator, discussed earlier in this chapter, if you want to render a large number of images for an
animation. Although the 12 hours required to render a single image may be acceptable on a once-a-week basis, it won’t work when you need to render 30 frames a week. Of course, you have this option only if your software supports hardware or network acceleration.

For more information on the topic of Macintosh hardware, see Chapter 5, “Hardware.”

The future

Among the makers of the many 3D applications currently available, no single developer has a stranglehold on the market—unlike other market segments such as photo retouching and illustration. The benefit of this wide-open market is that with no clear industry leader, each company has to work hard to improve its software.

On the down side, the 3D market is much smaller than the other markets, and probably will remain so. Whether such a small market can support all these companies remains to be seen. At one time, AT&T Graphics marketed MacTopas, a sophisticated 3D modeling and rendering program. AT&T has since let the product revert to the original developers (who now sell the product). AT&T may have decided to abandon the 3D Mac market to pursue other interests, or they may have found it unprofitable (they were selling the product for $7,000!).

For all of that, 3D applications continue to be important tools for multimedia developers. As the processing power of the Macintosh improves, you can expect the performance of these programs to improve, as well.

The next chapter discusses QuickTime, a topic of particular interest to those creating 3D animations.
Chapter 10

QuickTime
What is QuickTime? The simple answer to that question is that QuickTime is software, developed by Apple, that supports time-based media on the Macintosh. An example of time-based media is video—a sequence of images that are displayed onscreen to create the illusion of motion. The images are time-based because to retain synchronization with a sound track, the images must be displayed at the correct time in relation to the sound track.¹

But there's much more to QuickTime than just maintaining synchronization between sound and video tracks. And QuickTime isn't just video—QuickTime can also be used for animation or even text. A full discussion of QuickTime could fill one or two other books. The discussion in this chapter introduces you to QuickTime software and the ways it can benefit you in preparing your multimedia presentations.

**TIP:**

I'm told that the *QuickTime Handbook* (Hayden, 1992) is a very good book about QuickTime for those who want to go into the subject in more depth.

This chapter begins with an explanation of how QuickTime works and why it's important. If you already are familiar with QuickTime, you may want to meet the rest of us later in the chapter. The section "Installing QuickTime" tells you how to install the QuickTime extension on your Macintosh and play QuickTime movies. The "Recording Video" section explains the basics of how to record video using QuickTime on a Macintosh. The remaining sections introduce some video digitizing boards, accelerators, and software products you may consider using in your work with QuickTime.
Understanding the importance of QuickTime

QuickTime adds support for time-based media to the basic Macintosh system architecture. The most common media that require this kind of support are video, animation, and sound. Sound has been supported on the Macintosh since its introduction, but it's important that QuickTime supports sound because QuickTime can synchronize sound tracks with video tracks. Similarly, there have been applications that supported animation since the early days of the Macintosh, but while these applications usually let the animator adjust the rate at which animations are played, and even add sound to animations, none of the programs closely synchronized these elements. The most common example of this problem is that animations would play at different speeds when played on different machines.

QuickTime provides this synchronization. Most of this chapter centers on QuickTime's application to video, but remember that much of this discussion also applies to sound and animation.

QuickTime stores video digitally—that is, it converts the frames of video to digital images (computer representations of the image) that are stored on the computer's hard disk. When the video is "played," these images are read from the disk and displayed on the computer screen. The advantage of storing the video this way is that the computer can very quickly and easily access any part of the video sequence, just as it can quickly and easily access other information on a hard disk. Another advantage is that once the frames are stored on the hard disk, they can be manipulated using a number of different editing applications (which are discussed at the end of this chapter). The disadvantage of storing video this way is that the computer has to do a lot of work to display the video sequences—so much work that most Macintosh models can only display comparatively small movies—small both in size, about 240 x 180 pixels, and frame rate, around 15 frames per second (fps). This is much smaller than video seen on television sets which is 640 x 480 pixels and 30 fps. It is possible to improve this performance by adding hardware acceleration (see the section later in this chapter).
Onscreen Video

You don't have to store video sequences on your hard disk to play video on a computer screen. It is perfectly possible to take an analog video signal (most video signals you will encounter—from VCRs to cable television—are analog) and display them on the screen. To do this you will need a source (such as a laser disc player) and a video display board for the computer that will display the video images on the computer monitor.

The signal goes into the display board, which adds them to the graphics being output from the computer, and displays the results on the screen. The computer itself is only peripherally (forgive the pun) involved in the video; although the video image is displayed by the video card, the computer processor does nothing except tell the display board where on the screen to add the video to the graphics the computer is generating.
Even without digital video, being able to play video on a computer monitor is quite an achievement. If you have a laser disc player you can control the player from the computer, and create a sophisticated presentation. But this achievement is not as exciting or as useful as digital video (storing the video on the computer). Interacting with analog video from an external source is much more difficult. This chapter devotes its discussion to the topic of true digital video.

What’s so neat about digital video?

It’s easy to understand the importance of synchronizing time-based events. You probably can even understand why people want to synchronize animation and sound on the Macintosh, but what’s so neat—or useful—about these tiny windows playing video on the Macintosh screen? Well, nothing—and everything—is neat about digital video. Digital video today is often unimpressive because using it on the computer is very difficult (as you see later in this chapter). Digital video files are huge in comparison to the average graphics and text files, and the image quality and the frame rate of the video are usually very low.

Digital video, however, has several benefits for the multimedia producer or user. First, it is new, and computer people get very excited when they can do new things with their computers. Second, after the digital video is in your computer, you can edit and manipulate the material, and distribute it to others, with relative ease. Digital video also is very easy to add to multimedia presentations (the process is much easier than hooking up your VCR and trying to cue things from there). Finally, as you will see later in this chapter, the addition of special hardware makes it possible to play video that almost rivals what you see on your television set.

But, to be honest, at the moment we all look forward to the day when we easily can record and play back television-quality video on a
computer—in other words, the day when we can watch Saturday 
Night Live on our computer (now there would be a giant step forward 
in multimedia!). When that happens, why will you need your TV? And 
how will you tell the difference between your TV and computer? 
These questions present some interesting philosophical issues, but 
we leave them to the big thinkers and get back to digital video and 
QuickTime.

What’s the difference between animation and video?

Wait a minute, you’re thinking, what’s the difference between anima-
tion and digital video? In truth, the two aren’t much different. The 
Saturday morning cartoons on television are good examples of how 
animation can be video and video can be animated.

Subtle differences between animation and video emerge, however, 
when you work with your computer. Animation programs have been 
available for the Macintosh since its inception. Digital video, on the 
other hand, has only been around for a few years. The big difference 
between animation and digital video is the amount of data each 
requires the computer to deal with. Animation is much easier to store 
and play on a computer than is video, because animation files tend to 
be simpler and smaller than digital video files. Even if your animation 
uses very low frame rates, viewers probably won’t tell you, “That 
doesn’t look very lifelike.” The background may be one static graphic, 
and the animated character a sequence of seven or eight small 
images that repeat as the character moves across the screen. This 
arrangement saves memory—and also means that not much changes 
onscreen from one frame to another.

Video’s makeup is very different. The frame rates have to be higher 
or movement looks unrealistic—we expect more of “real-life” images 
than we do of animated characters! More importantly, even if one 
video frame appears to be very similar to another, they may be 
completely different—a slight move in the camera changes the colors 
of all the pixels in a frame. Figure 10.1 illustrates the difference 
between sequences of animated frames and video frames.
How does QuickTime work?

To achieve digital video the computer needs two things. First, the computer must be capable of handling video's large amounts of data. Typically, computers accomplish this by streaming (in streaming, the computer displays one frame as it reads the following frame from the disk). Second, the computer needs some kind of timing mechanism that keeps video and sound synchronized. Timing is particularly important when you display people speaking, because without synchronization their lips won't move in synch with the sound.

QuickTime fulfills both needs. QuickTime has many features, but the two most important are its compressors and its timing mechanism. The compressors enable the program to reduce the frames to a manageable size, so the computer can stream the information from the disk. The timing mechanism ensures that multiple tracks maintain synchronization.

Without compression, the large amount of data would prevent even the fastest hard disk from playing a movie—by the time the computer had read the first frame from the disk, the time for displaying the frame would have passed. Even with compression, QuickTime may not be capable of playing all the frames in the movie—for example, a Quadra is capable of capturing a much larger movie than a slower machine, such as an LC, can play. In theory, the timing mechanism in QuickTime handles this problem by dropping frames. A 20-frame movie might play at only 10 frames per second on an LC, in which
case QuickTime plays only every second frame in the movie. QuickTime always tries to maintain the quality of the sound, since any loss or skip in sound is very noticeable.

The algorithms that provide compression are not as important as the fact that QuickTime provides support for compression. Although QuickTime comes with several different compression algorithms (called compressors), you can add other compressors or even use hardware-based compressors to improve the performance of QuickTime movies.

QuickTime involves much more than the information covered here—this section has breezed over the details to tell you about the important things. In Chapter 7, "Graphics," you can read about JPEG—an application of QuickTime compression that is useful for working with still images. You can learn more about QuickTime by picking up a copy of either Cool Mac QuickTime or the QuickTime Handbook, both published by Hayden.

### Installing QuickTime and playing movies

QuickTime is a single file, called QuickTime. You must place this file in the System Folder (the Extensions folder, if you are running System 7) and then restart the computer to add the QuickTime functionality to your system (see figure 10.2).

![QuickTime extension with its Help balloon](image)

**Figure 10.2.** The QuickTime extension with its Help balloon.

At this writing, QuickTime only runs on a color-capable Macintosh—for example, the Macintosh II models, the LC models, the SE/30, most
PowerBooks, and all Centris and Quadra models. QuickTime does not run on a Macintosh Plus, Classic, SE, or PowerBook 100. The list of computers upon which QuickTime runs may change with the release of QuickDraw GX (a new version of the graphic routines that every Macintosh uses to draw graphics on the screen).

Your computer must be running System 6.0.7 or greater. The Sound Manager (part of the operating System) changed with version 6.0.7 and QuickTime requires the new Sound Manager. Further, if you run System 6, and have a pre-Mac IIci computer, you must install Color QuickDraw. Color QuickDraw is on the Printer disk of the System disks that came with your Macintosh, and must be installed manually. Better yet, install System 7 and don't worry about it, because Color QuickDraw is built into System 7.

When you restart the Macintosh, the QuickTime icon appears onscreen during the startup sequence. The icon is the only visible evidence that QuickTime is installed, unless you have an application that requires QuickTime to run, or you have a movie and an application that plays QuickTime movies.

**TIP:**

To play a movie with QuickTime, you must have an application that is capable of playing QuickTime movies. You need something like Movie Player (released with QuickTime and distributed by some user groups and bulletin boards), Popcorn (a shareware program distributed by user groups and bulletin boards), or one of several utility players that have been released with applications.

If you don't have a player application, but must see the movie, you can play it—but you have to restart your computer to do so. Name the movie StartupMovie, place it in your System Folder, then restart your system. The movie should play during the start-up. This technique is recommended only for emergencies!
Recording Video

After you have installed QuickTime and played a few movies, you may want to start recording your own movies. To record movies you need items included in the following list (and shown in figure 10.3):

1. A video source: this is the video your Macintosh will digitize. The source can be a VCR, camcorder, or LaserDisc player.

2. A video digitizing board: usually installed inside the computer, the video digitizing board converts the analog video signal to a digital image. The most common boards are NuBus cards that plug into one of the slots in the Mac II, Centris, and Quadra class computers. Another video digitizing board is a version of SuperMac's VideoSpigot that fits the LC's PDS slot.

3. Cables: use these to connect the video digitizing board to the source. Cables are very important! RCA-style or the newer S-Video cables are most commonly used.

4. Digitizing application: this application works with QuickTime and the digitizing board to capture a sequence of images to a QuickTime movie. No digitizing application comes with QuickTime (though the application Movie Recorder is included with the QuickTime Starter Kit sold by Apple). Most of the video digitizing boards now available come with some application for capturing video to QuickTime.

5. Sound digitizer: an optional feature. If you want to record sound, you must digitize the sound (see also Chapter 11, "Sound"). The video digitizer may include a sound digitizer (such as the Radius VideoVision) or it may not (the SuperMac VideoSpigot does not). If no sound digitizer is included, then you can use the built-in sound recording hardware (if your Macintosh has it) or you can use a MacRecorder (also described in Chapter 11).
Figure 10.3. Everything you need to record QuickTime movies.

TIP:
If you want to record from television, record the material to tape first and then digitize the material from the video tape—you then have more than one chance to get it right. Also, you must use a VCR or a television with a monitor-out option to record from television. Very few digitizing boards have a built-in television tuner, and therefore cannot directly capture a signal coming from, for example, your cable network.

The compressor’s role in digitizing

The frame rate of the movies you capture depends primarily upon the speed of the computer you use. The biggest bottleneck is the time it takes to compress an image, though the time it takes for the hard disk to store the information can affect performance. Your choice of compressor also affects the movie’s frame rate.
Generally, when we talk about **compressing** a movie we refer to either digitizing a movie using the compressor, or recompressing a movie that has already been captured. When we talk about **decompressing** a movie, we refer to the process involved in playing the movie back on the computer.

QuickTime comes with several compressors; some are better for still images and others are better for movies. QuickTime's current compressors are None, Video, Photo JPEG, Graphics, Animation, and Compact Video. These compressors are discussed in more detail later in this chapter.

**TIP:**
None? What’s a None compressor, you ask? The None compressor does no compression; it just passes the information out to be stored on the hard disk. Because the None compressor doesn’t compress, it can sometimes save the frames to disk much faster than any of the other compressors. The resulting movies are very large, but have higher frame rates. You can recompress such a movie with another compressor (usually the Video or Compact Video compressor) to reduce the size of the movies.

On the disk is a movie (Movie Test) and a still image (Image Test) that have been compressed using each of the compressors. Examine these movies and images to see the different sizes and performance that each compressor provides.

The most commonly used compressor for video clips is the Video Compressor—a good general-purpose video compressor. The advantage of this compressor over the Compact Video compressor described later in this section is that the Video Compressor is almost symmetrical (see the following Note).

**NOTE:**
A **symmetrical** compressor takes about the same time to compress a movie as it does to decompress the movie. An
Asymmetrical compressor may take a lot longer to compress a movie than to decompress it.

Asymmetrical compressors compress movie frames more efficiently because they can spend more time examining each frame and finding the best way to compress it. Asymmetrical compressors make movies much smaller, but are slower and require an extra step because you must capture the movie first and then recompress it.

You can use a symmetrical compressor to capture a movie in real time and then play the movie back; you can use an asymmetrical compressor to play a movie back, but not to capture a movie. Of course, if a compressor takes 10 minutes to compress a frame and 10 minutes to decompress the same frame, the compressor is classed as symmetrical, but is not useful for playing the movie. As a rule, therefore, you can assume that a symmetrical compressor also is a fast compressor.

The following section describes the compressors included in QuickTime, and offers some recommendations for their use.

QuickTime compressors

As stated earlier, QuickTime comes with several compressors. Each of these compressors has its own strengths. The following information can help you select the compressor best suited to your individual task.

- **Photo JPEG:** This compressor reduces the size of photographic images dramatically, but the images retain exceptional detail (usually, you cannot detect the difference between the JPEG-compressed image and the original). The high compression factors make it useful for reducing the size of photographic images, but the JPEG compressor takes several seconds to decompress large images.

- **Graphics:** This compressor is optimized for still images created using an 8-bit palette. It is not suitable for video.
• **Animation:** This compressor works best with computer-generated animation. It does not work well with video because of the "noise" inherent in video sequences (*noise* describes the way pixels in video images change color subtly from frame to frame, even though the photographed objects remain the same color).

• **Compact Video:** This compressor is extremely efficient for compressing video sequences, but is very slow. You can compensate for this slowness by using the None or Video compressor to capture sequences, and then use the Compact Video compressor to recompress the sequences. Compact Video-compressed movies typically are half the size of movies compressed using the Video compressor, but visually are still of high quality.

### Some tips for compressing video clips

As a QuickTime user, you are likely to want to compress video clips. You can experiment to determine which compressor best accomplishes this task on your computer.

To begin, try capturing a clip from a videotape or disc, so that you can capture the same clip two times; capture the clip once with the Video compressor and once with the None compressor. When you capture with the None compressor, you must recompress the clip afterward (okay, you don't have to, but uncompressed clips are very large and disk space is still expensive).

See which compressor provides the best performance during the capturing phase. The None compressor may provide the highest frame rate. If you have a fast computer, however, you may prefer using the Video compressor directly. The advantage of recording with the Video compressor is that you don't have to mess around with recompression. Only if you are going to give the clips to others, press them on a CD, or have serious space problems, do you need to recompress the clips using the Compact Video compressor.
Several applications that capture video—DiVA VideoShop, Apple’s Movie Recorder (on the QuickTime Starter Kit), and Adobe Premiere—provide an option that automatically records using the None compressor and then immediately post-compresses using the compressor of your choice. This option saves you the hassle of manual recompression, but doesn’t save a tremendous amount of time.

Selecting and using digitizing hardware

In order to record movies you must install a video digitizing board in your computer. The board converts the analog video signal to a digital image. The following sections describe some of the video digitizing boards currently available for Macintosh users, and explore the issue of hardware acceleration.

Some video digitizing boards

You can use the following overview as an information resource when you select your video digitizing board. The overview covers only some of the digitizing boards currently on the market, but can show you what features to look for in selecting the right board for your system.

SuperMac’s VideoSpigot

This low-cost board became tremendously popular soon after its release. VideoSpigot’s low cost and reasonably high quality of captured sequences makes the board popular with both beginners and seasoned professionals. The board’s initial popularity was boosted by the fact that the first VideoSpigot boards included a copy of Adobe Premier at no extra cost!
Included with the VideoSpigot is a special application called Screen Play, which uses SuperMac's own compressor to capture movies (see figure 10.4). This application provides much better performance than any of the built-in compressors, and captures at much higher frame rates than most other boards can obtain. A llci, for example, captures at about 12 frames per second (fps) with a RasterOps 24STV, and over 20 fps with a VideoSpigot.

Figure 10.4. The SuperMac VideoSpigot ScreenPlay Application.

The VideoSpigot board comes in two versions: a NuBus board for regular Mac II-class machine, and an LC version that plugs into the PDS slot of the LC. Those users with one foot in the Windows world can purchase a PC version of the VideoSpigot.

If you are considering buying a VideoSpigot, be aware that although the board is very good for small frame capture (anything less than 1/4 screen, 320 pixels x 240 pixels), it does not perform as well with larger frames. This limitation becomes an issue only if you want to capture large single frames from video tape—otherwise, you are unlikely to notice a difference in your movies. Another limitation of the VideoSpigot board is that it has limited controls for brightness and contrast. These controls are particularly important when your source material is poorly or unevenly lit. Further, the VideoSpigot sometimes puts black pixels in extremely bright (saturated) areas when recording from tape. Some people have encountered this
problem and others haven't—the source material seems to be the determining factor.

Finally, the VideoSpigot only has an RCA video plug in, so if you have a VCR with S-Video out (either Hi-8 or S-VHS) you cannot take advantage of the higher quality signal possible with an S-Video connection.

**RasterOps boards**

RasterOps offers a family of digitizing products with different features at different prices. The base member of the family is the 24STV. The 24STV is a video digitizing board combined with a video display board. If you connect the 24STV to your computer monitor, it displays video in 1, 2, 4, 8, and 24 bits; that means the board displays in two, four, 16, 256, or millions of colors. The 24STV does not support 16 bits (thousands of colors). For most applications, lack of 16-bit support is not a problem. To capture video, however, you must set the board to 24-bit mode. In 24-bit mode, a frame of video is larger than the same frame in 16 bits. This is because in 24-bit mode, the color of each pixel is represented by 24 bits while 16-bit uses 16 bits. So a 24-bit frame is half again as large as a 16-bit frame. This increase in the amount of data that has to be passed from the digitizer to the computer results in a smaller number of frames that can be transferred in a given period of time.

A IIici can capture at the rate of about 12 fps with a frame size of 180 x 160 using the 24STV board. Although the 24STV frame-rate performance isn't equal to that of the VideoSpigot, the quality of its captures is much higher. The improved quality is particularly noticeable if you are capturing large frames. Also, the 24STV brightness and contrast adjustments are much wider than VideoSpigot's.

The 24STV performs best with either a very fast Macintosh, or in non-real time capturing (see the following Note). The 24STV can display a window with video in it—when you want to display video from a laser disc during a presentation, this board is ideal.

Included with the 24STV is the MediaGrabber application, which you can use to capture still images and QuickTime (see figure 10.5).
NOTE:

When a compressor is unable to compress and save a frame before the next frame in the video must be compressed, you can use one of two methods to capture in non-real time. Both methods require special software.

Multi-pass capturing plays the tape at the usual speed and captures as many frames as possible in a single pass, then performs additional passes until all frames are captured. As an example, if the compressor can only capture one in six frames per pass, the compressor captures frames 1, 7, 13, and so on in the first pass; the tape rewinds, the compressor captures frames 2, 8, 14, and so on. The process repeats until all frames are captured. The advantage of multi-pass capture is that it doesn't place any extra stress on the video tape. The disadvantage of this method is that it requires a professional VCR with time-code support in order for QuickTime to know which frames have been captured.
Step-and-grab capturing places the VCR into pause mode, grabs the frame, then steps the VCR forward to the next frame and grabs that frame. This method places more stress on the video tape and requires a VCR with a clean pause. The method does not require time-code support, however, so you can use a less-expensive VCR.

The 24STV is a good general-purpose board, but you also may consider the RasterOps MediaTime board (see figure 10.6). The MediaTime board has the same video circuitry as the 24STV, but adds the audio circuitry from an AudioMedia audio board and provides support for CD-quality audio. For a little extra cash, you can get the 24MXTV. Its functionality is similar to that of the 24STV, but the 24MXTV also has graphics acceleration and its display circuitry works in 16 bits as well as 24 bits. Capturing in 16 bits accelerates the movie capture, because the amount of data being captured is smaller.

Figure 10.6. The RasterOps MediaTime board.

Radius VideoVision

The Radius VideoVision costs twice as much as the RasterOps 24STV, but offers more features. The VideoVision has a built-in 8-bits sound
digitizer (see Chapter 11, "Sound"). The Radius VideoVision enables sound mixing, so you can mix sound from the Mac and from an external source. The board has a built-in video encoder to enable you to record to video tape, and it has a function called \textit{convolution}—a special filter that improves the quality of Macintosh graphics when recorded on video tape.

As a video display and video digitizing card, the Radius VideoVision rates about the same as the 24STV, but has slightly better quality. Capture rates are about the same in both boards. If you need the video output capability, consider the VideoVision; otherwise, you might want to go with a cheaper alternative. (But read the following section on hardware acceleration before making your final decision.)

\section*{Hardware acceleration}

Before leaving the issue of digitizing hardware, this chapter must mention the important issue of hardware acceleration. As you can see from the preceding discussion, most existing digitizing boards, coupled with QuickTime, are capable of only very low frame rates at small frame sizes. How are you going to see \textit{Saturday Night Live} at full-screen size and at high frame rates?

The current answer to this problem is the use of hardware acceleration of the compression algorithms. This involves adding a plug-in board that is specially designed to run compression algorithms and can increase the capture rate and frame rate.

SuperMac has released a digitizing hardware acceleration product called Digital Film, and RasterOps has a similar product called MoviePak. Digital Film is unrelated to SuperMac's VideoSpigot discussed previously, but the MoviePak board is a plug-in daughter board that plugs into several of the RasterOps boards, including the 24STV, MediaTime, and 24MXTV. Radius has announced a plan to release a daughter board for their VideoVision product.

The benefit of these systems is that they increase the frame size and frame rate of the movies you can play; but these systems are not perfect—yet.
These boards aren't capable of true 640 x 480 frame size playing at 30 fps. The SuperMac and RasterOps products are compressing at either 640 x 240 or 320 x 240 pixels per frame. Further, you must have a Quadra to get good performance from these boards—they require a lot of horsepower and fast hard disks to work properly. The movie files are very large, and a second of video can be between 20 and 60 megabytes! Currently, the quality of movies produced with these boards barely rivals the quality of VHS video, so the boards aren't suitable for broadcast video applications. Another disadvantage is that you can't play these movies without the hardware boards, and the boards are expensive.

As a rule, you need hardware compression only if you want an off-line editing system, or if you want to capture large frame-rate and frame-size movies, reduce the size and frame rate, and compress them with the Compact Video compressor.

**Using QuickTime movies**

There's more to QuickTime than just capturing video to a QuickTime movie. Once a movie has been created, it can be played in a number of applications. There are also several tools available for editing QuickTime movies.

**Play them**

There are a number of applications that play QuickTime movies, and most developers are adding support for QuickTime to their applications where it is appropriate. These programs include Director (see Chapter 14, "Authoring environments," for more information on Director), Action! (an interactive presentation program from Macromedia, described in Chapter 15, "Interactive presentations"), Special Delivery (an interactive presentation program from ITC, also described in Chapter 15), and Persuasion from Aldus, to name just a few. Okay, let's not forget MovieWorks, Cinemation, and that flashiest of all programs, Microsoft Word! Also, Apple has Movie Player, an application for playing movies that is on the Apple QuickTime Starter Kit.
Edit them

Editing is the fun part! Several applications enable you to edit QuickTime movies. Some of the programs mentioned above supply some editing capabilities. For example, MoviePlayer—the player application included on Apple's QuickTime Starter Kit—enables you to select a range of frames in a movie, cut or copy the range, and then paste them into another movie or somewhere else in the same movie (see figure 10.7).

![MoviePlayer interface](image)

**Figure 10.7.** Selecting a range of frames in MoviePlayer.

If you want to do more sophisticated things, such as adding special effects, transitions, or titles, you need to investigate one of the QuickTime editing applications.

**QuickTime references**

Before proceeding further, we need to talk briefly about references. QuickTime movies can be large. Even with compression, a typical movie clip is larger than one megabyte; copying and pasting files of that size can become difficult, impossible, or just serious trouble. Apple came up with a solution. When you select part of a movie
and copy that movie clip to the Clipboard, the system copies only the first frame in the clip sequence and some pointers to the selected data in the original file. The pointers act like an index to the clip.

When you paste the clip into another file, the system only inserts the pointers to the original file—if you paste into a program that doesn’t support QuickTime, you get only the first frame of the sequence.

As an example, when you make changes to a movie in MoviePlayer and then select Save As, a Save dialog with extra options at the bottom appears. These options enable you to save the movie as a reference file (which still contains the pointers to the original file) or as a stand-alone movie (which copies the information from the original file into the new file). This figure shows how references work.
Using references saves disk space, because it prevents multiple repetitions of information. If you give the new file to someone else, however, you also must give them a copy of any referenced file or they cannot play the movie!

Some programs use references, and some don't; you must be on the lookout for how a program works. And not all programs display the references used in a movie the way Movie Player does. When you create a new movie using editing programs such as VideoShop and Premiere, the programs usually give you the options of using references or creating a standalone movie. Others give you no options—you must experiment with the program or read the manual to determine if the program enables references.

Using programs that support QuickTime

This section presents a brief overview of some of the hundreds of programs that support QuickTime. The overview doesn't cover all the programs that can play QuickTime movies (like Microsoft Word, WordPerfect, or Persuasion), because that information is too lengthy for this chapter, and the individual details of these programs are not very interesting.
Utilities

Two screen capture utilities are available that record activities on the screen as QuickTime movies and then let you play them back. The programs, Spectator from Baseline Publishing, and CameraMan from Vision Software, offer similar functionality. Both can capture either the whole screen or part of the screen to a movie, and both enable you to add sound afterward by recording a narration as the movie plays. Figure 10.8 shows the CameraMan Settings screen and its options.

CameraMan and Spectator are useful for creating training movies, but have many other uses, as well. One of the more imaginative applications of these programs I have seen used PageMaker to manipulate a scanned image and recorded the process in CameraMan.

When using these programs, be aware that attempting to capture a full-sized screen in color results in very low frame rates—perhaps only 4 fps or less. Reducing the area to be captured improves the performance. Alternatively, working in 1-bit color rather than 8-bit (if you can) makes much smaller movies. If you use 1-bit color, you may be able to fit a one-minute movie on a single floppy disk.

Both programs support sound. They will add sound effects representing mouse-clicks, as well as let you record a narration to go with the movie. Figure 10.9 shows sound in CameraMan.

Included on the CD-ROM is an application called MoviePlay, which is supplied with CameraMan. MoviePlay is a basic QuickTime movie-playing application.
Editing

If you get seriously interested in creating and working with QuickTime movies, you may benefit from buying a QuickTime editing application. Two such editing programs are currently available: Adobe’s Premiere 2.0, and DIVA’s VideoShop 2.0 (see figures 10.10 and 10.11). With either of these programs you can edit QuickTime movies, apply effects, add transitions, and add titles.

The two programs seem to target slightly different markets. Premiere aims at the high-end editing market, and VideoShop at the general market. Premiere supports time code, for example, and VideoShop requires a plug-in external module to enable time-code support.

These editing applications have other differences you should consider. Premiere simulates (in a Preview window) the playback of the movie you are editing; VideoShop generates the effect as you add it to a movie, so you can see the effect immediately. What does this mean? When you add an effect in VideoShop, you must wait a few seconds or minutes while VideoShop creates the frames that make up the transition. Adding an effect to the score in Premiere is quicker, because it requires no processing time to generate the effect. Premiere can’t play the effect for you when you preview the movie,
however, because it hasn't performed the calculations necessary to generate the final frames that make up the transition. Instead, Premiere simulates the effects in the Preview window. If you don't have a fast machine or you want to see exactly what a transition looks like, the Preview may not be good enough; if that is the case, you must create that portion of the movie anyway, and thereby negate Premiere's speed advantage.

![Figure 10.10. The Premiere editing application.](image)

Figure 10.10. The Premiere editing application.

![Figure 10.11. The VideoShop editing application.](image)

Figure 10.11. The VideoShop editing application.
If you are just starting out and want a low-cost package, you may be interested in a low-cost editing program named Sparrow that DiVA has recently released (see figure 10.12). Sparrow offers many of the features found in VideoShop, but at a much lower price. Interestingly, DiVA was purchased in early 1993 by Avid, who makes a line of professional editing systems that run on the Macintosh.

![Figure 10.12. The Sparrow editor from DiVA.](image)

**Post-processing tools**

Post-processing tools are advanced tools that really are intended for adding special effects to clips. VideoFusion (see figure 10.13) and After Effects are two such programs that work with QuickTime. They both offer a large collection of effects, many of which aren’t currently available in VideoShop or Premiere. Both Premiere and VideoShop offer some effects, however, and most people need to get an editing program before they get something for processing clips; therefore, check out the basic programs and see what you really need before making a selection.

**Other utilities**

Some additional utilities do interesting things with QuickTime movies. TheatreMaker creates a standalone application that plays a QuickTime movie. You can specify a background image upon which
the movie plays. The program comes with several backgrounds, including a theater, TV set, and a drive-in movie screen; it's a fun way to send a movie to a friend.

Figure 10.13. VideoFusion.

Shutdown Movie is a companion to Apple's built-in StartupMovie feature. Shutdown Movie plays a movie when you shut down your Macintosh.

Included on the disk are copies of TheatreMaker and Shutdown Movie.

Cross-platform applications

This chapter cannot end before mentioning that Apple has released QuickTime for the PC. QuickTime for Windows plays QuickTime movies created on the Macintosh on a PC running Windows. Although QuickTime for Windows currently doesn't enable you to do all the Macintosh editing functions, future versions of the program may offer more editing capabilities.
Microsoft has released a product called Video for Windows that provides many of the features of QuickTime. You can use Video for Windows to create and edit video (on the PC); the product even includes a converter that converts QuickTime movies to the Video for Windows format.

It makes you wonder: if it weren't for Apple, where would PC developers get ideas for new products?

**The future**

Apple is enthusiastic about QuickTime and continues to improve the product. You can expect to see new compressors and hardware support (both from Apple and third parties). You also can look for the addition of support for interactivity within movies—enabling you to jump to another part of a movie by clicking within the movie, for example. New computers with built-in support for QuickTime acceleration are likely to appear. And, you can expect the frame size and frame rate of movies you can record and play to improve until they rival television. With these advances, the difference between a computer and a TV/VCR will diminish—and we may get less and less work done, as well.

But that's all in the future, and until then, it's time to get on to the next chapter and worry about what we hear when we watch video. *Saturday Night Live* might be fun to watch, but you need sound to hear the punch line.
Figure 10.14. Saturday Night Live on my computer.
Chapter 11
Sound
Perhaps the most overlooked aspect of multimedia is sound. Audio. What you hear. It’s so easy to get wrapped up in animation or video on the screen that you completely overlook the quality of the sound that you are listening to.

And yet the Macintosh has a tremendous advantage over the millions of PCs out there. Every Macintosh comes with a small speaker and the System software to play sound. It means that you can have sound in any multimedia presentation or game without having to worry whether the user will be able to hear it. Short of a malfunctioning speaker (which has been known to happen), the sound will play.¹

### Storing sound

It’s important to know a little about how sound is stored and used on the Macintosh—its limitations and how it can be improved.

### Macintosh built-in sound hardware

The hardware built into current Macintosh models supports 8-bit sound sampled at 22 kilohertz (kHz). These two numbers have a direct relationship to the quality of the sound that a Macintosh can play. Remember that computers work with numbers. They’re digital. Sound is analog—a continuous sound wave. To “record” a sound, the computer measures the level of the sound and converts it to a number. This process is repeated many times a second, and these numbers are then used to recreate the sound during playback. The 22 kHz represents the frequency (how often) a sample is taken. Hertz means “per second,” so 5 hertz would be five times a second, and 22 kilohertz is 22,000 samples per second (which seems like a lot, but you will soon see that it’s not as good as it could be). The 8 bits is the accuracy with which the sample is stored. With 8 bits, a computer can store 256 different values, so from the lowest to the highest point in the signal there are only 256 steps. As a comparison, CD quality sound, which is 16-bit sound, has 65,536 values from the lowest to the highest signal. Figure 11.1 shows how the analog wave form is sampled and converted to digital information.
Figure 11.1. The Macintosh represents analog sound waves in a digital form.
Macintosh sound quality

An 8-bit sound at 22 kilohertz doesn't sound too bad when played through the internal speaker of your Macintosh, but it will sound even better if you listen using head phones, or if you plug external speakers into the headphone jack at the back of the Macintosh. Of course, even with these additions, the 8-bit, 22 kHz sound won't rival the quality of an audio compact disc (CD), which uses 16 bits and samples at 44.1 kilohertz.

The Macintosh internal hardware is not capable of playing sounds at rates higher than 8 bits, 22 kilohertz. You can record at lower sampling rates (5, 7, and 11 kilohertz being the other choices). These are all still 8 bits, but the lower sampling frequency means that the number of samples is smaller and the size of the sound file becomes smaller. Of course, the quality of the sound decreases, too.

Lower sample rates

The biggest reason for sampling at lower rates is that the size of the sound file is reduced. A ten-second sound at 22 kHz may take up 224K, but the same sound at 11 kHz is 116K, and at 7 kHz it is 77K. This is particularly important if you want to use a lot of sound and distribute it on a floppy disk. But you also have to consider the quality of the sound that the listener will hear. As the sampling rate goes down, the quality of the sound deteriorates, too.

For most productions you should use either 22 kHz or 11 kHz. I frequently use 11 kHz because the difference between 22 kHz and 11 kHz will not be noticed by most listeners when they are using the Macintosh internal speaker. They will notice the difference if listening through headphones or playing external loudspeakers. If the sounds are soft, with a lot of quiet passages, then 22 kHz will sound much better than 11 kHz. The other two sample rates (7 and 5 kHz) introduce a lot of noise and should be used only in desperation or for a particular effect. Some people think 5 kHz sampling sounds like a telephone conversation. I think it sounds a lot worse, but that's for you to decide.
The CD-ROM contains a demo program called Sound Quality that contains the same sound sampled at different qualities. You can play the sounds and decide which sound quality you prefer.

**Sound compression**

There is a form of sound compression called MACE (Macintosh Audio Compression and Expansion) that you can use to make your sounds smaller. The samples are made at 22 kHz, but the values are compressed when they are stored in the file, and then decompressed during playback. The compression scheme is not "lossless," that is, the quality of the sound is not completely preserved. Also, some applications do not support compressed sound format.

Compressed sounds are smaller than 22 kHz sounds, but then so are sounds sampled at lower rates (11 or 7 kHz.) You should try compression options to see whether you prefer them when compared to sounds sampled at lower sampling rates. I prefer 11 kHz sound to the sound compressed at the minimum compression factor, which results in sound files just a little smaller than the sound file at 11 kHz.

**Recording sound**

Until the Mac LC and Ilsi were released, the Macintosh did not include sound digitizing hardware. You could play sound with your Mac, but if you wanted to record sound, you had to buy an external piece of hardware. The Mac LC and Ilsi changed that. They included hardware for digitizing sound at 8 bits, and they also came with a microphone. Since then, all of the new Macs have come with this functionality built-in, so you can start recording sound with a Mac straight out of the box. Figure 11.2 shows the microphone included with the Mac LC and Ilsi.
If you have an older Macintosh, you can still buy an external piece of hardware to enable you to record sound. You just need to buy a MacRecorder (made by Macromedia, and sold for around $240 mail order). The MacRecorder is a box that attaches to the modem or printer port of your computer and adds the hardware necessary to record sounds. MacRecorder also comes with an excellent application called Sound Edit, which you can use for recording and editing sound (more on that later). Figure 11.3 shows a MacRecorder. Figure 11.4 shows Sound Edit.

Upgrade to System 7.0

If you aren’t currently running System 7.0 then you really, really, really should consider upgrading. Yes, you will need to have at least four megabytes of memory, but memory is fairly cheap, and System 7.0 provides a lot of neat features (not the least of which are that you can change the icons of folders and applications easily, and you can play sounds by double-clicking them)!

Perhaps a more important reason is that more and more applications will run only in System 7.0, or at least, certain functions do not work unless System 7.0 is running.
If you absolutely, positively, can't run System 7.0 just yet, then at least make sure you are running System 6.0.7 or 6.0.8. Apple updated the Sound Manager (part of the operating system) when it introduced System 6.0.7, and several things, including QuickTime, do not work under earlier versions of the system.

Figure 11.3. MacRecorder box.

Figure 11.4. The sound editing application Sound Edit, included with Mac Recorder.
If you only want to record new beep sounds for your computer to play, and you have a microphone or MacRecorder, then you can record the beep sounds in the Sound control panel. Along the bottom of the Sound control panel you see a list of the sound recording hardware devices attached to your computer. If your Macintosh has a built-in microphone, you will see an icon representing the microphone. If you have a MacRecorder, then you see the MacRecorder icon. Figure 11.5 shows the Sound control panel with the built-in microphone option. Figure 11.6 shows the Sound control panel with the MacRecorder icon.

Figure 11.5. The Sound control panel shows the built-in microphone option.

Figure 11.6. The Sound control panel enables you to choose an input sound device (in this case the MacRecorder). Clicking the Options button shows which serial port the MacRecorder is connected to.
Select the icon (if it is not already selected) and then click the Add... button. A window appears for recording sounds. To record a sound, click the button labeled Record and then click the Stop button to end the recording. Click the button labeled Play to review the sound. Figure 11.7 shows the Record Sound dialog box. If you like the sound, save it by clicking the Save button; otherwise record another sound or click Cancel to close this window.

When you save a sound, the sound is stored in the System file, and you can use it as a new beep sound by selecting it in the Alert Sounds field of the Sounds control panel.

![Figure 11.7. The Record Sound dialog.](image)

**Sampling versus Digitizing**

When recording sound with a computer, audio people like to refer to the process as "sampling" (recording samples of the audio). But the process is also a conversion from analog to digital, so computer people often use the term "digitizing." Is it sound sampling or sound digitizing, and does it really matter? As long as you know what you mean, and understand when someone uses the other term, I don't think it matters which term you use.

You may need to obtain the MacRecorder driver in order to use the MacRecorder with the Sound control panel. The MacRecorder driver is a small piece of software that lets the System communicate with the MacRecorder hardware. You don't need the driver if you are using Sound Edit, but it is necessary if you want to record sound in the Sound control panel, use the Audio palette in HyperCard, or record sound in QuickTime. The MacRecorder driver is included with later versions of the MacRecorder, but if you have an "old" version of the MacRecorder, the driver may not have been included. Contact Macromedia for a copy of the driver.
Recording sounds using the Sound control panel is interesting, but you will soon wish that you could edit the sounds in some way. If that's the case, and you don't want to spend a lot of money, then you might consider using the Audio Control panel included with HyperCard. This is a set of XCMDs (see the chapter on HyperCard if you don't know what an XCMD is) that enable you to record sounds, perform simple editing, and save sounds in HyperCard stacks. It's useful, too, if you actually want to use the sounds in HyperCard. Figure 11.8 shows the HyperCard sound palette, and figure 11.9 shows the HyperCard sound editing window, available by clicking the Edit button on the sound palette.

Figure 11.8. The HyperCard audio palette enables you to record and edit sounds.

The more adventurous should definitely get a copy of either Sound Edit (previously mentioned) or AudioShop from Opcode. Even if you have a Macintosh with a built-in microphone, you should consider getting one of these programs because they allow you to edit sounds and apply effects such as echo and reverb.
One of the most confusing things about sound is the large number of file formats that sound can be saved in. The information stored in the file formats is essentially the same (that is, the sound data), but you really need to know about these formats. When you get a sound file in one format, you may not be able to use it in a particular application.

The most common sound format on the Macintosh is a sound resource, which has a file type of 'snd.' In the Macintosh file system, a file can contain either data or resources. Resources are chunks of information stored inside a data file that are accessed separately from the data stored in the file. The resources are accessed using the Resource Manager, which is part of the operating system. It's kind of like a mini-database (don't tell Apple I said that; one of the things Apple tells programmers not to do is use the Resource Manager as a database).

Programs like Sound Edit let you save files as resources within files. HyperCard works with sounds only in this format. You can store several sounds as separate resources inside a HyperCard stack, and
the stack will play the sounds. To play a sound, simply create a button with the script Play 'sound name'.

Another file format that has become a standard is the Sound Edit file. This is the native file format of the Sound Edit application, and because Sound Edit was practically the only application for recording sounds for years on the Macintosh, this format has become fairly well known. Many programs can read files in the Sound Edit format.

AIFF is a relatively new sound format that has become popular. AIFF stands for Audio Interchange File Format, and there are two important features of this file format. First, you can transfer sounds in this format to computers other than the Macintosh. More important to multimedia developers and users is that, unlike sound resources, an AIFF file can be played from your hard disk. (Sound resources have to be loaded entirely into memory before they can be played. This not only requires time to load the sound, but it also means that you are limited by the amount of memory you have available—if you don't have enough memory, you can't play the sound). On the down side, playing sound from disk does mean that the disk is being accessed, and this can cause problems if you are trying to read something else from the hard disk at the same time!

There are some shareware utility programs available that you might find useful for working with sound files.

Sound RoundUp by Robert Gibson is a drag-and-drop utility that makes a copy of all the 'snd' sound resources in a selected file and copies them into a folder or suitcase that it creates. Sound RoundUp is capable of scanning entire directories and disks.

Sound Converter Pro by David Lambert converts sounds from one sound format to another. Sound Converter is useful for converting sounds that aren't in the "right" format for your application.

Both Sound RoundUp and Sound Converter are included on the disk.
Using advanced sound techniques

There's more to recording good sound than connecting a microphone and clicking the record button. The following tips will help you do a better job.

Getting the best sound at 11 kHz

If you decide to use sounds recorded at 11 kHz, and you are using a program like Sound Edit, then you should record the sound first at 22 kHz, and then copy the sound across to an empty 11 kHz file. Sound Edit will convert the sound to 11 kHz, and the sound is generally a little better in quality than what you would get if the sound had been sampled at 11 kHz directly. This is because the program takes two samples and averages them in the conversion process. This produces a sample that is more accurate than the single sample made when digitizing at 11 kHz.

Use a tape recorder

For best sound quality, record a sound with a tape recorder first and then digitize the sound from the recording. Do this for two reasons. First, the quality of the microphone included with the Mac, and in the MacRecorder, is not that good. Second, tape is much cheaper and more plentiful than computer memory, so you can record several takes and then record just the right one.

To record from a tape recorder to the Macintosh built-in hardware, you will need a cable and an adapter. The cable was included with the IIsi and LC originally, but it is now no longer included, so you will have to make a trip to an electronics store to get the right parts. The Radio Shack part numbers are #42-2461 attenuating cable, and #275-375 adapter.
Watch out for clipping

When recording sound, you must ensure that the sound does not clip—that the sound level is not too high. Clipping results in a very distorted sound and is usually not pleasant. If clipping occurs, adjust the level and re-record. (That's another reason to use a tape recorder.)

Avoid recording too low

Just as important, don't record with a sound level too low—to hear the sound you'll have to amplify it and this will accentuate noise. (The quality of the sound is reduced because you are effectively lowering the range of sample used to represent the sound.)

So how do you get the level right? Some of the recording applications provide a visual read-out with a peak meter just like a traditional tape recorder. If the application provides some kind of meter, you want to adjust the level so that the loudest noise doesn't reach the maximum level of the meter. If there is no meter, then you will have to experiment to see what sounds the best.

Get a good original for better sound

Although you can achieve a lot of effects using programs such as Sound Edit and AudioShop, you are still limited by the quality of the original sound. Although post-processing can turn a good sound into a great sound, it's generally not possible to turn a poor sound into even an average sound. You must concentrate on getting the best quality original rather than plan on fixing it in the mix later on.
Using advanced hardware

Although the Macintosh internal hardware is limited to 8 bits and 22 kHz, that doesn't mean that you are limited by the same constraint. You can buy a NuBus card that offers much better sound quality. DigiDesign makes the AudioMedia II card for around $1,000. This board is capable of recording at 16 bit, 44.1 kHz, which is CD quality audio. The AudioMedia board provides great quality sound as well as tools to work with the sound files. You can even buy an application called DECK from DigiDesign that has an interface resembling a four-track tape recorder; it enables you to record and mix sounds.

You will need a lot of disk space to record sounds, and a reasonably fast hard disk as well. And if you give someone a copy of your presentation, they must have the AudioMedia card to play back the sound at high quality.

Using CD

If you have a CD player attached to your computer, you are probably wondering how to integrate the sound from an audio CD into a presentation. Until recently you couldn't! You could put an audio CD into the CD-ROM drive, and most drives would recognize the disc and even let you “play” the disc (through an external amplifier). But you couldn't copy the files to your hard disk and then open them and edit them, or play them. This is because the audio CD uses a very different format for storing sound from that used by any Macintosh application.

The latest version of QuickTime, version 1.6, has changed that. Provided you have one of the new generation CD drives (the Apple CD300 or 300i) you can open CD audio tracks and convert them into QuickTime movies. You can then use these files in any application that supports QuickTime format movies.
You can get some XCMDs that will “play” a track from an audio CD when you tell it to. But the sound won’t come out of the Macintosh internal speaker—you must have headphones or an external speaker connected to the CD player. The only way around this is to redigitize (resample?) the sound by connecting the CD player to your sound digitizer and record the sound in Macintosh format. Be sure to have permission from the record company to do this (see also Chapter 14, “Clip media and copyright”).

Using MIDI

This chapter is almost over, and I haven’t said a word about MIDI (Musical Instrument Digital Interface). MIDI is a method of communicating with electronic musical instruments (for example, electronic keyboards). MIDI tells an instrument which notes to play and for how long, and also other information such as how loudly to play the notes. It’s a very compact way of sending sound information from one instrument to another, or from the computer to an instrument and back again.

There are many applications available from companies such as Opcode and Passport that enable you to communicate with a MIDI device using the Macintosh. These are often called MIDI sequencers, and with them you can play a composition on an instrument and record the MIDI information on the computer. You can edit the file (fix any mistakes you made) and then play back the new sound by sending it back to the instrument. For the computer to be able to communicate with a MIDI instrument, you need a device that plugs into the Macintosh serial port and converts the signals coming from the port into a MIDI signal. How the computer communicates with the instrument is shown in figure 11.10.

Because of the compressed nature of MIDI files, and the fact that all the computer has to do is send the information out through the serial port, the computer doesn’t have to do a lot of processing. It can be doing other interesting things, such as playing an animation. Several applications support MIDI playback, including Macromedia’s Director and Passport Producer. Neither program provides sophisticated editing of MIDI files, so you will need a recording and editing application in addition to these programs. Figure 11.11 shows Metro, from OSC, a MIDI sequencer.
Figure 11.10. How the Macintosh communicates with an electronic instrument.

Figure 11.11. Metro, a MIDI sequencer, enables you to record, play, and edit MIDI files.

Finally, though MIDI has several advantages, it does require additional hardware (the instrument and MIDI box) to play the sound. MIDI cannot be used for digitized sound (such as voices). For these reasons, MIDI is really suitable only for some applications.
Metro is an example of a MIDI sequencer that is currently available. Even if you don't have a MIDI keyboard, this program will play a sound through the internal speaker. This is not a great way to play sound, but you can at least play with the program that way. A save-disabled copy of Metro is included on the disc.

Making the Macintosh speak

In computer-synthesized speech the computer constructs what sounds like a human voice based on English text provided to the synthesizer. It has been available on many computers for several years now. The quality of even the best talking programs does not rival the quality of the human voice, and for many applications digitized narration sounds much better. But for some applications a synthesized voice makes more sense. Reading mail or reading text files that come from a database that changes continually are examples.

Unfortunately, there has been only one application available for making the Macintosh speak. This program is called MacInTalk, and it was written for Apple back when the Macintosh was first introduced. Unfortunately, while Apple still distributes the program, Apple does not own the source code to MacInTalk, and for that reason the program has not been updated since its introduction (although it has been patched so that it continues to operate with newer versions of the operating system).

There is even an XCMD for HyperCard that works with MacInTalk, and the program is available on bulletin boards, from user groups, and from APDA. It is an unsupported product, however, and there are no guarantees that the program will work with future versions of the operating system. For these reasons, I would recommend against using MacInTalk for any purposes.

There is hope, however. Apple has shown a new version of MacInTalk that is being developed with more voices and improved sound quality. It is anticipated that this new version will be released sometime soon, at which point using computer-synthesized speech for multimedia productions may be worth considering.
Looking ahead

Apple released in the middle of 1993 a new version of the Sound Manager that supports 16-bit sound.\(^2\) At about the same time, Apple will release new multimedia computers that use a built-in DSP (digital signal processor). A DSP is a separate processor that is specifically designed to handle digital signals such as those generated by the analog-to-digital and digital-to-analog conversions of sound digitizing. These separate processors are required because of the amounts of data being handled when playing 16-bit audio at CD data rates. If the main processor of the Macintosh were used for this process, the computer would not be able to do anything else while the sound was playing. The AudioMedia board (described above) uses a DSP chip to provide 16-bit sound on existing Macintoshes.

Providing a separate chip makes it possible to play sound while the Macintosh is doing other things, which is important to multimedia users and developers. It’s not all good news for multimedia, however. Although using a DSP increases the sound quality, it also means that the user has to have separate sound hardware (speakers) attached to the Macintosh to hear the better sound quality. Also, with the increased sound quality comes an increase in data rates. You will be able to play a QuickTime movie with 16-bit sound from a hard disk, but it’s unlikely that you will be able to do the same thing from a CD-ROM. Finally, the large installed base means that for the next two or three years, 8-bit sound will remain common for commercial products although 16-bit sound will become standard for kiosks, presentations, and other special applications.

This chapter completes the section on the basic data types of multimedia. The next chapter looks at the programs used to bring these different data types together into the final production.

\(^2\)There is a rumor about why it has taken Apple so long to produce a 16-bit version of the Sound Manager. Apple was sued by Apple Corporation (the Beatles’ recording company) over sound hardware. Although Apple Corporation had allowed Apple to use the name Apple while selling computers, it didn’t want Apple to be operating in the sound market (and cause confusion over which company was which). The rumor was that Apple agreed not to release sound hardware (such as keyboards) and not to update its computers to support 16-bit sound.

A more likely scenario is that Apple realized that until it released computers that supported 16-bit sound, it didn’t make sense to expend resources on a new version of the Sound Manager.
Chapter 12

Clip media and copyright
Whether you are a graphic designer, electronic artist, or multimedia producer, you may on occasion lack the time or skill to create the material necessary for a production. Perhaps you haven't the time to fly out to San Francisco and photograph the San Francisco-Oakland Bay Bridge, or you haven't the equipment to scan in images of different textures. For these and many other occasions, the best answer may be to obtain production material from clip media collections or stock houses.

Clip media collections originally began with the advent of desktop publishing. Several companies started selling collections of images—called clip art—that could be used for desktop publishing. Now companies are selling clip media collections—collections of sound, images, animations, and QuickTime movies. Stock houses were in existence before clip art. Used by advertising agencies and corporate graphics departments, stock houses assemble collections of photographs and movie and video clips that they license for use.

This chapter discusses some production material sources and issues of copyright and rights of ownership—in particular the necessity for getting releases from models—that apply to multimedia.

**Stock houses**

Stock houses and film libraries are the traditional sources for photographs and movie clips. These repositories of visual media have served the advertising and television industries for some time. Many of the generic images seen in magazines (a mountain, a running stream) are available from stock houses. Film libraries provide access to a large number of historic film clips, as well as generic video; the Empire State Building, or a busy street. Rarely are images such as clips from movies or news images available from these kind of sources—you must go to the holders of the copyright for that kind of material.

As a multimedia producer, you can obtain production material from stock houses and film libraries. However, many stock houses have
little or no experience with multimedia, and may be reluctant to license their materials for such use. Further, their rates may be high; most photo stock houses are used to dealing with advertising agencies and charge accordingly. Also, they will search and find a clip that matches your needs, rather than sell you a collection of material that may or may not be useful to you. The search and individual licensing costs more. Check the classified advertising sections of magazines like New Media and Macworld for possible sources for this material.

Video Tape Library Ltd.
Video Tape Stock Footage
1509 North Crescent Heights Blvd., Suite 2
Los Angeles, CA 90046
(213) 656-4330

Jasmine Stock Video/Music
(310) 277-7523

Clip media

Several multimedia clip collections are currently available. Most clip collections are distributed on CD-ROM, because compact discs can store a large quantity of material and are inexpensive to produce. These collections often include photographs, graphics, sound, animation, and QuickTime clips—all on a single compact disc. Collections sell from $99 to $399, depending upon the source and the type of material.

The following paragraphs discuss some of the clip media collections currently on the market.

- *ClipMedia* includes sounds, animation, and background images. The image shown in the figure is from a catalog created for the ClipMedia disc using Aldus Fetch (a catalog program sold separately and designed for cataloging multimedia clip collections). Figure 12.1 shows some of the material contained on the ClipMedia disc. The index shown in the figure was generated
using Aldus Fetch, a program that is useful for indexing a large collection of media (see the section on Media cataloging utilities). The ClipMedia collection is available in both Mac and PC versions from:

Macromedia
600 Townsend Street
San Francisco, CA 94103
(415) 442-0200

![Image of ClipMedia CD Source Catalog]

**Figure 12.1.** Some of the material in the ClipMedia collection.

- **Multiple Media Tour** is a collection of various multimedia materials from Audio Visual Group. The Multiple Media Tour CD-ROM includes its own catalog of the source material in the collection (see figure 12.2). The catalog is in six languages, including Japanese. Multiple Media Tour is available from:

Audio Visual Group
398 Columbus Ave, Suite 355
Boston, MA 02115-6008
(617) 381-1609
Figure 12.2. A portion of the Multiple Media Tour catalog of source material.

Other clip collections include:

Clip Time
Images for presentations.
Alpha Technologies Group, Inc.
6921 Cable Drive, Suite 100
Marriottsville, MD 21104
(301) 781-4200

Digital Video Library
Video clips of cities, places, and sports.
Educorp
7434 Trade Street
San Diego, CA 92121-2410
(800) 843-9497 (US and Canada)
(619) 536-9999 (international)

WraptureReels One
Video footage, animation, and audio tracks for presentations.
Form & Function
San Francisco CA
(415) 664-4010
Multimedia Music
A diverse collection of music created for computerized multimedia productions.
Filler Tracks
6534 Sunset Boulevard
Hollywood, CA 90028
(800) 877-0078

Although some of these collections offer hundreds of different clips and may be very good values, to determine whether or not a collection saves you money you must consider how many of its clips you are likely to use. In some cases, you may need to use only a few clips to recoup the cost of the collection.

Media cataloging utilities

Whether you buy clip media discs or just start collecting your own material, you will soon accumulate a large collection of files, and finding the clip you need will become more difficult. A number of companies are now selling utilities for cataloging the many different files and media types. Nearly all of these programs produce a catalog of thumbnails (small graphic representations of the files) that you can view, as well as some kind of search function that will search by file name, file type, or other characteristic. Some of these programs are multi-user and work over a network, so they are particularly appropriate for businesses with large collections of material spread over several machines in the office.

Just two of the programs available are:

Aldus Fetch (illustrated in figure 12.1)
Aldus Corporation
411 First Avenue South
Seattle, WA 98104-2871
(206) 622-5500

Media Cataloger
Interactive Media Corporation
P.O. Box 0089
Los Altos, CA 94023-0089
(415) 948-0745
Copyright laws

If you are creating a production for commercial distribution, or if you are using works created by others, you need to be aware of copyright laws and their application to multimedia.

Copyright and your work

Copyright laws extend to multimedia productions. You are wise to obtain a copyright for any multimedia production you create, even if you have no intention of selling your work to others. A copyright, to some extent, protects your work (or a derivative of your work) from being copied and sold by others. A copyright protects work for the life of the work’s creator plus 50 years.

Obtaining Copyright for your work

Gaining copyright is very simple. Under current copyright law, a work is protected by copyright as soon as the work is created. You must indicate, however, that the work is protected. To indicate the copyright, place the following notation on the work in an obvious location:

© Year Name of author

For example,

© 1993 Michael D. Murie

Some minor variations within the notation are possible, and a sound work should have the ® symbol instead of the © symbol. This symbol indicates that the work is a sound recording that cannot be perceived visually.

Although you are not required to register your work with the Copyright Office, you may benefit from doing so. If someone copies your work, you must register before you can file a suit against them. Further, you receive a reduced amount of damages if your work is unregistered at the time the infringement takes place. If you anticipate any possibility of future infringement on your work, register the work with the Copyright Office.
Copyright and Work for Hire

Although copyright law assumes that a work's creator holds the copyright for that work, this assumption does not hold true in the case of Work for Hire. If a work is created as a Work for Hire, the work is the property of—the person who hired the creator. If you hire me to create an animation for you under a Work for Hire agreement, for example, you own the work and its copyright. Both artists and those who hire them need to be clear on who will own the copyright to works produced under employment agreements.

A Work for Hire must be designated as such in a contractual agreement. If the agreement does not state that the work is a Work for Hire, the work is the property of the creator. The creator can assign the copyright to someone else, but this assignment requires extra paper work.

The employer can claim to be the creative force behind the creator's work, and therefore the rightful owner of the work's copyright. Say, for example, that you hire me to take a photograph; you tell me where to stand and where to point the camera, and you arrange the subjects of the photograph. Because you directed me in almost every aspect of this photograph's creation, you have a strong claim to the ownership of the photograph's copyright. Although you may win the copyright to this work, you must do so in court. The safest arrangement, therefore, is to specify who is to be the owner of a work before the project begins.

Copyright and the work of others

At the same time, as a multimedia producer you must be aware of the rights of others. You cannot take another's work and use it to create a derivative work unless you obtain the permission of the artist who created the original. Copyright law expressly grants to the owner of
the original work the right to make copies and derivatives. Some famous cases have set the precedent for this law, such as that of the sculptor who made a sculpture that was based on a photographer's published photograph. The court judged the sculpture to be a derivative work.

Some exceptions exist to the copyright limits on the use of the works of others. These exceptions fall under the Fair Use section of copyright law. Fair Use grants to others the use of protected works under certain conditions, such as for criticism, research, and satire. Fair Use also permits the reproduction of an original work as an element of another work, such as in the creation of a montage. The law stipulates, however, that the reproduction(s) cannot be a major portion of the new creation.

One factor in determining the fair use of another's work is whether the use may detract from the original creator's attempts to sell the work now or in the future.

You should be very careful about using material under the fair use clause for any commercial or public use. For anything other than a prototype, you should either gain permission to use protected material, buy material from a clip media collection or stock house, or create your own new material.

Releases

When your productions include photographs or other likenesses of individuals, you must get a release from the individuals depicted in those likenesses. A release is a written agreement that states that an individual grants to you (the multimedia producer) the right to use that individual's likeness in any way you please. Remember that the release must specify that you can use the likeness in any way; otherwise, your agreement is open to interpretation.

Obtaining a release is particularly important when your work is to be distributed commercially.
The future

Ted Nelson, who coined the term “Hypertext,” has spent the last decade promoting Xanadu, a world-wide electronic publishing medium. The idea behind Xanadu is that it serves as a single system for publishing materials. Users can access the material, though they may have to pay a charge. The user can copy and excerpt material, use the material to create new works, and then publish these new works on the network. When another user accesses the new work, the system calculates the royalties for the use of the new work. A percentage of the royalties goes to the creator of the original work, and a percentage goes to the creator of the new work.

Sound complicated? Sure does, and maybe that’s why no such system currently exists. The closest system is AMIX, an electronic network where people can “publish” material and others can buy the work; AMIX and the original author split the royalty. AMIX has no provision, however, for derivative works the way the Xanadu system would.

More electronic networks for clip art, sounds, and even movies, are likely to appear in the future. There are at least two services under development for distributing photographs.

End users will benefit greatly from electronic networks of publishable materials. Not only will the networks bring down the price of multimedia clip art, but they will speed the work of multimedia producers. At last, you will be able to quickly find and get a copy of that perfect background for your slide presentation!

The explosion in electronic publishing poses a challenge for the creative industry, because digital media is so easy to copy and distribute. Much work remains to be done in educating end users about what is permissible when working with the creations of others.
Part IV
Multimedia environments
Chapter 13

Choosing the right development environment
A development environment is the glue that combines different media elements—images, sounds, text, and animation—into a coherent project that the user can explore. Although programs such as Photoshop and Infini-D enable you to create bit-mapped graphics or three-dimensional illustrations, they don’t enable you to add buttons or data fields. Similarly, though Premiere is a sophisticated tool for editing QuickTime movies, QuickTime does not enable the user to do anything except play the movie from beginning to end and stop and start it manually.¹ This really limits the productions you create, and that’s why you will nearly always want to have access to some kind of development environment such as HyperCard, SuperCard, Director, or Special Delivery. These programs are described in more depth in Chapter 14, “Authoring environments,” and Chapter 15, “Interactive presentations.”

The different environments

Even though the first part of this book has devoted considerable attention to Director and HyperCard, they are not the only multimedia development environments available. They are two of the most flexible development environments, but with this flexibility comes added complexity. Other tools may be better suited to your application. The tools available for multimedia development can be divided into three categories:

1. Slide presentation environments
2. Interactive presentation environments
3. Authoring environments

Figure 13.1 shows how these environments relate to one another, to QuickTime, and to programming an application from scratch.
Slide presentation programs such as Persuasion from Aldus and PowerPoint from Microsoft are based on the traditional slide presentation metaphor. You create a series of screens that are linked together in a sequence, and you show them in that order. Though it's possible to jump from one slide to any other slide, it's usually expected that the presentation will proceed in a certain sequence.

Many of the principles used in creating a sequential presentation are the same as those used in creating an interactive presentation, but there are some specific things to remember when creating a presentation that will be projected or used with a speech. These things include avoiding type that is too small to read when projected and limiting the number of points per slide. These hints are just the "tip of the iceberg" on this subject; there are numerous books devoted to creating presentations.
PowerPoint or Persuasion may be suitable for your application. They don’t, however, offer the functionality and flexibility of the other programs discussed here, so they will not be covered in any greater depth. If you do use PowerPoint or Persuasion, you will still use many of the other tools described in Part 3, “Tools.”

Passport Producer is a hybrid application for creating multimedia productions. In some respects Producer fails the interactivity requirement, because the only interactive option provided is the ability to make a presentation pause until a user clicks. However, Producer does offer several other functions that make it of interest to multimedia producers.

A presentation in Producer is made up of several tracks. Each track can hold either a sound, a QuickTime movie, a graphic, text, or a MIDI file. You can synchronize several things by arranging them side by side in separate tracks at the same point in time. Figure 13.2 shows a presentation arranged in Producer. Other features include the ability to pause in a presentation and a standalone player that can be used to distribute productions created with Producer. With these features, Producer almost makes it into the interactive presentation category.

Figure 13.2. Passport Producer.
Interactive presentations

The interactive presentation programs enable the multimedia designer to add buttons that when clicked take the user to another location in the presentation. Most of these programs have some things in common with the slide presentation programs. They often use the concept of slides or screens, and you can add transitions from one screen to another.

The addition of buttons that cause different things to happen, such as the playing of a QuickTime movie, or a transition to one of several other screens, makes it possible to create much richer multimedia productions. Both Action! from Macromedia and Special Delivery from ITM offer buttons and linking features. There are no scripting functions, but you can still create quite complicated presentations. These programs are described in some depth in Chapter 15, "Interactive Presentations."

Authoring programs

Programs such as HyperCard, Director, and SuperCard add another element to the multimedia developer's arsenal—scripting. With scripting it is possible to create very complex interactions between the user and the presentation. For example, by clicking a single button, the user might inquire about a subject and then find all the information available on that subject in the presentation.

These authoring environments can be used to create slide presentations or simple interactive presentations, but what sets them apart from the other classes of programs is the ability to write scripts and create complex interfaces.

Just because HyperCard, SuperCard, and Director are in the same category, they don't have the same strengths and weaknesses, nor should they even necessarily be used to create the same kinds of multimedia presentations. Their differences are explained in more depth in Chapter 14, "Authoring environments."
Custom programming

It is possible that none of the commercial programs available will be able to perform the task you want to accomplish. In that case you may have to go to custom programming. Though custom programming makes it possible to do almost anything imaginable, it is much more expensive and time consuming than any of the commercial applications already mentioned.

It is beyond the scope of this book to discuss this avenue of development in more depth. If you are considering this route, you should find someone with experience in Macintosh software design and development to provide advice about the feasibility and cost of your desired project.

QuickTime

You are probably wondering where QuickTime fits into the world of multimedia development depicted in figure 13.1. As you have probably noticed, QuickTime is off on its own with a circle pointing to each of the other areas. That's because QuickTime is not really a development environment like the others. It more closely resembles a data type, just as PICT is a data type for graphics, and AIFF is a data type for sound.

Though QuickTime can be used to create movies and may even replace the slide presentation programs for some jobs, it is far more likely that QuickTime will be used to provide animation or video from within one of the development environments.

A future version of QuickTime may provide options for interactivity, in which case it may be possible to use QuickTime in place of both of these other environments. This possibility depends upon the actual functionality provided when and if QuickTime supports interactivity. Even if interactive support is added to QuickTime, you will still need an application to define the interactivity options and create the QuickTime movie. QuickTime would be merely a distribution medium.
The decision

Many factors go into choosing a development environment, not the least of which are cost, ease of use, and features. Each of the programs has different strengths and weaknesses, making it even more difficult to choose among them.

Here is one recommendation, however. If you are interested in multimedia development, then you should consider beginning with either HyperCard or SuperCard. They are both inexpensive and can be used to create a wide range of presentations. Either would be a good introduction to the world of multimedia development. Director is great if you are interested in animation, but if all you need are buttons and data fields, then Director is too much—in terms of both cost and complexity.

Action!, Special Delivery, and Producer are worth considering if you don’t have any complex scripting needs. Reading Chapter 15, “Interactive presentations,” may help you to decide among the three programs.

The next chapter is devoted to an in-depth description of the authoring programs HyperCard, SuperCard, and Director. If you aren’t interested in these programs, you should skip ahead to Chapter 15, “Interactive presentations.”
chapter 14

Authoring environments
The authoring environments HyperCard, SuperCard, and Director provide the multimedia producer with a great deal of freedom. You can use these programs to create just about any type of production. They provide a great deal of power and flexibility, and yet are much easier to learn than a traditional programming environment or language such as C or Pascal. All three of these environments feature some kind of programming language—HyperCard and SuperCard use a language called HyperTalk, while Director uses a language called Lingo. These languages provide the power to create complex multimedia presentations and interfaces, featuring buttons, fields, and other interface components that are familiar to the Macintosh user.

Although you can use these authoring tools for nearly every production task, they aren’t necessarily the best tools for every job. You may find that one of the interactive presentation tools (described in the next chapter) can do exactly what you want and requires less effort than using the tools described in this chapter. For that reason, study both chapters and then decide which tool is right for you. You may need to know how to use several tools in order to create the different productions on which you work.

HyperCard—what is it?

Although HyperCard was introduced by Apple back in 1987, it remains unfamiliar to many Macintosh users. Is HyperCard an address book, a database, a hypertext environment, or something else? In some respects, HyperCard is all and none of these things. The best description of HyperCard I have heard comes from the program’s creator, Bill Atkinson, who describes it as a “do-it-yourself construction set”—a simple environment for creating just about anything you want.

HyperCard has been used to create address books, children’s games, electronic books, demos, and help systems. The following sections describe HyperCard, to help give you a better idea of how it works and what it does.

1Hypertext is a term that originated to describe documents that had links between words to other references for the word. For example, if you were reading somewhere else in this book and saw a reference to hypertext, clicking on the word would take you to this description.
How does it work?

The HyperCard authoring environment consists of an application (HyperCard) and the data files that HyperCard works with, called stacks. In many ways a stack is to HyperCard what a text file is to Microsoft Word—it's the document that HyperCard works with to do things. HyperCard's capabilities are very different, however, than those of a program such as Microsoft Word.

When you start HyperCard and open a new stack, you see an empty window (see figure 14.1). This window is your view of the stack's contents. The Tools palette, also shown in figure 14.1, provides a number of tools that you can use to manipulate the stack. With the exception of the top three (you learn about those later in the chapter), all of the tools are paint tools, similar to those you might see in a program such as MacPaint or Canvas. You can select a tool and start drawing on the stack. You can draw a picture, for example.

Figure 14.1. An empty HyperCard stack and the Tools palette.

HyperCard doesn't limit you to one window—you don't have to open another stack to create another illustration. A stack contains cards, which are screenfuls of information. To create a new card, you simply
choose New Card from the Edit menu. When you choose New Card, the stack screen becomes blank. Don’t worry, what you drew on the first screen—the contents of the first card—still exists in the stack. To go back to the original card simply choose Prev from the Go menu. The Next command in the Go menu returns you to the new card that you created.

Figure 14.2 shows the relationship of cards to the stack. Cards have a very specific order; card two always follows card one, unless you delete one of the cards or insert a new card between two cards. A stack can contain an almost unlimited number of cards. The capability for creating multiple cards provides a great deal of power to HyperCard’s users.

![Figure 14.2. The relationship of HyperCard’s cards to a stack.](image)

**Fields**

HyperCard would have limited use for many applications if it didn’t enable you to store information—text—in the cards. The Text tool, shown in the Tool palette illustrated in figure 14.4, lets you create
text, but it only creates bitmapped text; after you write the text it is converted into a graphic made up of a pattern of dots on the screen—just like something painted on the screen in a paint program. You can erase part of the text using the eraser tool (see figure 14.3), because the computer does not recognize the text as information. For this reason, you cannot search or change the text. To store text in a format that you easily can change and work with, you must first create a field to contain the text. A field is a container for text information.

![Bitmapped text with eraser tool](image)

**Figure 14.3.** Bitmapped text that has been partly erased with the Eraser tool.

To create a field, select the Field tool, shown in figure 14.4, then place the tool over the card and hold down the Command key; the cursor changes to a crosshair shape. Continue to hold down the Command key while you click the mouse button and drag out a rectangle; then release the mouse button. HyperCard creates a text field (see figure 14.5).

![Field tool](image)

**Figure 14.4.** The Browse, Button, and Field tools.
After you create a field, you can enter information into it. You must first switch back to the browse mode to enter information (you can move and resize the text field while the Field tool is selected, but you can’t enter information). To switch back to the browse mode, click on the Browse tool (see figure 14.4).

The field seems to disappear! Don’t worry, it’s still there. Move the cursor to the place where you drew the text field; the cursor changes from the hand shape to an I-beam. You can click and type some text into the field (see figure 14.6).

If you want to change the text’s font or change the field to a more visible position on the page, you can do so by editing the parameters.
for the field. To begin this procedure, switch back to the Field tool and double-click on the field. A dialog box appears that provides access to a number of parameters for the field (see figure 14.7).

![Field parameters dialog box](image)

**Figure 14.7.** Field parameters.

The Style options control the field's appearance. Transparent is the default style option. Any background or card graphics will be visible underneath a transparent field. When you use the Opaque option, the field obscures other fields or bit-mapped graphics over which you move the field. The Rectangle option draws a box around the field, and Shadow adds a drop shadow to the field. The Scrolling option turns the field into a scrolling field, into which you more easily can put a large amount of information. Figure 14.8 illustrates the effects produced with HyperCard's field options.

![Different types of fields in HyperCard](image)

**Figure 14.8.** The different types of fields in HyperCard.
The other options (Lock Text, Show Lines, and so on) all affect how you interact with the field and the format of the field's contents.

Clicking on the Font... button brings up a font dialog box that enables you to change the default font for the field (see figure 14.9). Clicking on the Script... button brings up a script window in which you can enter a script for the field. The next section of this chapter describes scripts in more detail.

![Figure 14.9. The font dialog box for the field.](image)

Finally, notice that the top of the field dialog contains a Field Name box (which is blank in the figure) and lists a Card field number and Card field ID (refer to figure 14.7). These elements are important in scripting because they are used to identify these objects in the scripts.

Set the field style to Rectangle and close the dialog box. Choose the Browse tool and you see that the field now has a rectangle around it. If you go to the second card in the stack (by choosing Next Card from the Go To menu) you see that the field does not appear on that card. The field you created is called a **card field** and exists only on the card in which you created it (see figure 14.10).
While a card field is useful for some applications, it is impractical for others. If you want to create an address book, for example, you will find it tiresome to recreate the fields in every card in a stack. To avoid such repetitive tasks, you can use HyperCard's backgrounds to duplicate common elements throughout several cards.

**Backgrounds**

You may not realize it, but the two cards that you created in the stack share a common parent, called a background. Anything you draw in the background of a card appears in all the cards attached to that background. To add something to the background of the cards in your stack, choose Background from the Go To menu. Notice when you do this that everything on the card disappears, and that the menu bar has a striped outline. Draw something in the background,
or type the word "Background" using the text tool and then choose Background again to switch back to the card; you now see both the contents of the card and the contents of the background. When you go to the next card, you see that it has the same background contents (see figure 14.11).

**Figure 14.11.** Three cards that share a common background.

If you add a field to a background, that field exists in all of the cards attached to that background. If you choose New Card from the Edit menu to add a new card, the new card is attached to the current background and displays the background's contents, even though the card contains nothing else.
What if you don't want the background to appear in a card? With HyperCard, you can have multiple backgrounds in a stack. To create a new card with a different background, choose New Background from the Edit menu. A new blank card, attached to a new background, appears. (HyperCard automatically creates a new card when you choose New Background, because every background must be attached to a card.) Figure 14.12 shows how different backgrounds can be attached to different cards in a single stack.

![Figure 14.12](image)

Figure 14.12. Four cards attached to two backgrounds in a single stack.

How do you know which background is attached to a card? Choosing the Bkgnd Info... command from the Objects menu displays the identification information of the background that is attached to the current card (see figure 14.13).
Figure 14.13. Identification information for the background attached to a card.

HyperCard attaches the background to a card when you create the card; you cannot change the background attached to an existing card (but you can delete the card). HyperCard attaches the background that is current when you choose New Card from the Edit menu. If you are in Background 1 when you choose New Card, HyperCard attaches the New Card to Background 1; if Background 2 is current when you choose New Card, HyperCard attaches Background 2 to the new card.

An important factor to remember about backgrounds is that a field created in a background exists in all the cards of that background. This arrangement is particularly useful if you are working with database-type applications; for example, if you are storing names and addresses in a stack.

You add a field to a background much the same way that you add a field to a card, except you add the background field while in the edit background mode (choose Background from the Edit menu).

**Buttons and scripts**

A **button** is an object that the user can click on to activate a response in the system. When the user clicks on a button, a sound might play, or the presentation might go to another card. Buttons are important, too, because they enable you to build up an interface.

Creating a button that takes you to another card is surprisingly easy. You begin by adding a button, using a process similar to that for
adding a field to a card. Choose the Button tool, the middle tool at the top of the tools palette, and draw a rectangle on a card (you can add a button to a background, but for this example, add the button to a card).

Double-click the mouse button and the button parameter dialog appears, as shown in figure 14.14. Choose Rectangle from the Style column, to outline the button. Type a name for the button in the button name field; for this example, type the name Go to next card. Next, click on the Show Name and Auto Highlight buttons. The Show Name option displays the name in the button; the Auto Highlight option highlights (displays in reverse tones) the button when the user clicks on it.

![Button parameters dialog](image)

**Figure 14.14.** The button parameters dialog.

You can choose to have HyperCard automatically link your newly created button to another card, so that when you click on the button HyperCard goes to the selected card. To create this link, click on the LinkTo button. A palette, shown in figure 14.15, appears with three options: This Card, This Stack, and Cancel. If the palette is in the way, you can move it to the side of the screen, but do not close or cancel the palette. Instead, use the Go To commands in the Go To menu to go to another card in the stack. When you are on another card, click on the This Card button; HyperCard creates a link to the current card from the button in the original card and then takes you back to the card that contains the button.
Let's try the button out. Choose the Browse tool and click on the Go To Next Card button. The button highlights and takes you to the linked card.

You may be wondering how HyperCard performs this action. Go back to the first card, either by pressing the escape key on your keyboard or by choosing the Previous command from the Go To menu. Select the Button tool and double-click the button on the card. When the button preferences dialog opens, click on the Script... button. The Script window opens (see figure 14.17).

The Script window contains a series of instructions that make up the button's script. The information in the script tells HyperCard what to do when the button is activated. The script shown in figure 14.16 reads as follows:

```
on mouseUp
  go to card id 4873
end mouseUp
```

Figure 14.15. The Link to palette.

Figure 14.16. The script window.
HyperTalk

HyperCard script is written in the programming language HyperTalk. HyperTalk is similar to the Pascal programming language, but is an English-like language.

Don't be afraid of the term **programming language**. You don't have to be a programmer to use HyperCard, you need only have a willingness to experiment. If you want to experiment a little with HyperTalk, try adding the line

```
visual effect iris open slow
```

just before the line, `go to card id 1234`. After you add the line, the script looks like this:

```
on mouseUp
    visual effect iris open slow
    go to card id 1234
end mouseUp
```

Now click the mouse button again, and you see a visual effect that happens as HyperCard transfers to the other card. You can select from a large collection of visual effects including dissolve, wipe, and checkerboard.

HyperTalk provides a wide range of other commands, this introduction barely scratches the surface. HyperTalk supports strings, variables, and control structures which enable you to do nearly all the things that sophisticated programming languages do.²

Before we leave HyperCard programming, it is important to briefly talk about how HyperCard knows to execute a script.
The script example we looked at began with the words `on mouseUp` and ended with the words `end mouseUp`. These script lines perform two very important functions. The words `on` and `end` indicate the beginning and the end of a script. You may want to store in the same button several scripts that perform different functions. You therefore must use the `on` and `end` indicators to tell HyperCard where each script begins and ends. The term `mouseUp` is the name of the routine. You must spell the name the same way at both the beginning and the end of the script. If the name is not identical in both places the formatting of the script is incorrect, and HyperCard will generate an error message when executing the script.

HyperCard spends a lot of time generating messages as the user does things like open a card, click on a button, or type in a field. HyperCard sends these messages to the scripts associated with the buttons, fields and cards. If a script exists with the same name as a message, then that script is executed.

For example, if you click on a button, when you release the mouse, the message `mouseUp` is sent to the button. If there is a script labelled `mouseUp` attached to the button, then that script will be executed. If there is no `mouseUp` script in the button, HyperCard sends the message to the card script, then the background script, then the stack script, and finally the Home stack script (see figure 14.17).

As a result of HyperCard's message sending capabilities, you could put a single script in the stack, and have that script do something whenever the user clicks on any button. (Although this arrangement is possible, it's not one you are likely to want.)

HyperCard sends many other messages, including `mouseDown`, `mouseWithin` and `mouseLeave`. Your button could contain scripts for all these actions:

```
on mouseWithin
on mouseDown
on mouseUp
on mouseLeave
```
Figure 14.17. Message sending in HyperCard.

**What is it good for?**

HyperCard can do so many things—from small databases to kids’ games—that describing its benefits can be difficult; an easier approach may be to describe HyperCard’s limitations. HyperCard does not replace a real programming environment. HyperCard has many limitations that make it impossible to create an application of any kind: the window size and number of windows is limited, the button types are limited, and the implementation of menus is difficult. HyperCard tends to be slow and does not support color. Although HyperCard supports text quite well, it is not appropriate for use with large database applications.
A major strength of HyperCard is that it is expandable. You can add additional functions to HyperCard by writing XCMDs and XFCNs. XCMDs and XFCNs are small programs that perform a specific function. You can copy and paste XCMDs and XFCNs into HyperCard stacks to add new functions, using either Apple's ResEdit or some other resource moving utility. QuickTime support and the display of color images on top of cards are just two of the functions that you can add with XCMDs.

Although you can write your own XCMDs, doing so requires some experience with Macintosh programming and is not for the faint-hearted. One exception to this rule exists: Heizer software sells a product called CompileIt! which takes HyperTalk script that you have written and tested in HyperCard, and compiles the script into an XCMD. The advantage of this arrangement is that, in many cases, the compiled XCMD executes much faster than the HyperTalk script executed by HyperCard.

**Where to go from here?**

This section only has scratched the surface of how to use HyperCard. A number of very good books on HyperCard are currently available that explain the details of HyperTalk programming in greater depth. You may want to buy one of those:

**HyperCard Stack Design Guideline,** Apple Computer. New York: Addison-Wesley, 1987. A bit outdated (it doesn’t cover the newer version 2.0) but does provide a lot of tips and suggestions for beginners.


Also, HyperCard comes with a very good electronic reference stack that explains nearly all of the HyperTalk commands, complete with examples (see figure 14.18). You can browse through that stack and use it as a reference.
Finally, stacks created by other users are a great reference. Many stacks are available from online sources and computer-user groups. You can get some of these stacks, take them apart, and examine the scripts, to find out how to create your own projects.

The future

The future of HyperCard looks much brighter now that Apple has decided to take it back from Claris and develop future versions of the software. Apple’s primary reason for reclaiming HyperCard is that they want to use it as a front-end to AppleScript, a new scripting language Apple is developing as a way of sending messages between applications. Apple’s renewed interest in HyperCard should ensure its existence for some time to come.

A major recent development was the release of the HyperCard Player. This allows the multimedia developer to add the HyperCard code to a stack to create a file that can be distributed to others. Even if these users don’t have a copy of HyperCard, the stack will play.

The following section discusses another authoring environment, SuperCard, that offers a similar but different range of tools for the multimedia producer.
SuperCard—what is it?

In its functions, SuperCard is very similar to HyperCard. SuperCard uses a language that is similar in syntax and shares many (but not all) of the commands used in HyperTalk. In fact, you can convert HyperCard stacks into SuperCard.

What is the difference between the two environments? The two most important differences are that SuperCard supports multiple windows of all kinds (plain windows, dialog boxes, palettes), and it also supports color. If you are working on a project that requires either of these functions, SuperCard is a natural choice over HyperCard.

On the down-side, SuperCard is slightly more complicated to work with than HyperCard, and is much slower, particularly if you are using color. Most users who need color, however, are willing to sacrifice some speed to get it.

How does it work?

Like HyperCard, SuperCard has documents (called projects) that are run by the SuperCard application. Whereas HyperCard has built-in editing functions, however, SuperCard has a separate application for document editing, called SuperEdit. You create and edit your project using SuperEdit, and you run the project using SuperCard. Some people find this separate editing application inconvenient, particularly during the debugging—making it work—stage of development.

SuperCard enables you to build a standalone application. In a standalone application, the code required to run the project is copied into the file, creating an application that can run on machines that don’t have SuperCard installed. The HyperCard Player, recently released by Claris, added a similar capability to HyperCard.

When SuperEdit opens a project, it displays a list of the project’s windows. A SuperEdit project window is like a stack in HyperCard—a stack provides a window in which cards and backgrounds exist. The relationship of cards and backgrounds to the window in SuperCard is the same as their relationship to a stack in HyperCard,
but in SuperCard, you can have multiple windows in the same project. Figure 14.19 shows SuperEdit being used to edit a simple project.

SuperEdit attaches scripts to cards or buttons in the same way that HyperCard does. Note in figure 14.19 that, unlike the tear-off palette in HyperCard, SuperCard tools are arranged on the left side of the edit window in SuperEdit.

When you use SuperCard, you have the option of locating scripts in the project. You can call such a script from any of the windows. This arrangement is similar to that of the Home stack in HyperCard. The scripts in SuperCard are very similar to those in HyperCard, as you can see in the example in figure 14.20.

Setting up menus is much easier in SuperCard because SuperEdit provides a separate list of menus. You call up the list by clicking on the Menus button in the project window. You can open and edit each menu item from a list and attach a script to the menu item that will be executed when the item is selected. Figure 14.21 shows where this is done.
What is it good for?

SuperCard's strengths lie in its multiple windows and color support. SuperCard is the tool to use if you require these capabilities. In nearly all other respects you can do exactly the same things with either HyperCard or SuperCard. Further, many of the add-on applications for HyperCard, such as CompileIt!, also work with SuperCard. Remember, however, that you sacrifice a bit of speed with SuperCard due to the added overhead required with color support. (The lag in performance is slight; most users won’t even notice it!)
Where to go from here?

Fewer support books and reference materials are available for SuperCard than for HyperCard. Due to the similarity of the two programs, however, you may find that HyperCard references and example stacks help you in scripting and using SuperCard. Even if you know HyperCard, however, you should refer to the manuals that come with SuperCard to familiarize yourself with the differences in syntax between the two programs.


The future

The future of SuperCard is unclear at this writing. SuperCard was originally published by Silicon Beach Software, but when that company was sold to Aldus, SuperCard didn't seem to fit with the interests of the new owners. Although Silicon Beach Software released a minor update (announced prior to the sale to Aldus), a year has gone by and SuperCard still does not support QuickTime. In early 1993, Aldus announced that they were interested in selling SuperCard, but the implications this announcement holds for the future of the product are unclear.

Although you can use SuperCard safely, if you are going to invest a lot of development in a project you should consider the future availability and support for the product before choosing SuperCard.

While SuperCard resembles HyperCard, Director is a very different kind of authoring tool.

Director—what is it?

Director began life as an animation program called VideoWorks for the Macintosh. For several years it was the only animation program
available, and for that reason it became very well known and has been used for all sorts of applications.

About the time HyperCard was released, a scripting language (called Lingo) was added to VideoWorks. Lingo is very similar to HyperTalk in syntax (but it’s not the same, which can cause confusion when working regularly in both environments).

Since then VideoWorks went through a name change, and the publisher—MacroMind—has changed its name to Macromedia. But the product is still very rooted in the world of animation.

Directors is somewhat difficult to learn and use, because its scripting language is added to what was originally an animation tool. Director remains very popular, however, principally because no other program does exactly what Director does. Macromedia has released frequent updates to keep the program current.

Also available is the Director Windows Player application. Player enables you to take a Director animation created on the Macintosh and play it on a Windows machine. You can play the animations, but not edit them using the Player on Windows. Most productions work without needing any changes.

At this writing, you cannot edit Director on a Windows machine, but a Windows version of Director is certain to become available eventually.

You can use a Director program called the Player to turn a Director animation into a standalone application that can be distributed to others. The Accelerator, another Director program, accelerates Director movies (makes them play faster). While the Accelerator has certain performance advantages, most people prefer to use QuickTime to accelerate Director animations.

How does it work?

The concept behind Director is very simple. Director has three major parts: the Stage (the window in which an animation plays); the Cast (a database of graphic images that are displayed on the Stage); and the Score (a spreadsheet that records when and where a cast member is visible on the Stage). These parts are shown in figure 14.22.
Director includes other elements (such as a Paint window for creating and editing the cast, and a Text window for entering text) but they only support the three major functions.

To create an animation, you must first create the objects that you want to animate. You can draw these objects in the Paint window, or copy them from another paint program and paste them directly into the Cast window. Figure 14.23 shows a cast member being edited in the Paint window.

Figure 14.23. The Cast window and Paint window.
After you have created the cast objects, it's time to arrange them on the Stage. To accomplish this task, simply click and drag individual cast members from the Cast window onto the Stage. If you look in the Score window, you see that the Score now contains a mark that indicates the cast member in the current frame. Figure 14.24 shows the cast member on the Stage and in the Score window.

Figure 14.24. A cast member on the Stage and in the Score.

In the Score window, the vertical columns indicate a single frame in the animation. The location of a single cast member is stored in one of the squares in the column. A cast member can be in any one of the horizontal rows (called a channel), but the order of their appearance indicates which cast member is drawn on top of another.

If you click in the next vertical column in the Score, the Stage becomes blank. The Stage is blank because you added the cast member to a single frame only. You can either drag the cast member to the Stage again, or select and copy the first block in the Score, and paste it into the next frame. If you copy and paste the cast member information and location, the location of the cast member is the same in both frames. To alter the location in the second frame, click and drag the cast member slightly away from the original location. Figure 14.25 shows a cast member that has been moved from one frame to the next. You continue this process to create the animation. You play the animation by clicking on the Play button at the top of the main Director screen.
Director stores Lingo scripts in the Score. Figure 14.26 shows an example of scripts in Director. You can attach the scripts to either a cast member or a frame. You can add commands to a frame; for example, you may want to add a command to go to a specific frame (to create a loop), or a command to pause the animation or to cause some other time-based event to occur. Director executes Scripts attached to cast members only when you click on the cast member. Attaching scripts to objects enables you to implement buttons and other user interface functions in Director.

When Director plays an animation, it calculates which cast members are in the frame and where, and draws each frame of the movie from the Stage and Cast information. The frames of the movie don’t exist
until you run it. This method of storing and playing animation differs
from that used by QuickTime or even the Director Accelerator. These
programs store all the frames of the movie in sequence on the disk.
When you play a movie back in QuickTime or the Director Acceler-
tor, the programs read the frames from the disk and display them
sequentially on the disk.

Director, on the other hand, loads all the Cast window members into
memory and then starts playing them based on the information in the
Stage window. The advantages of this method of animation are that
the movies are much easier to edit, and the movies are generally
much smaller on the disk than QuickTime movies (because Director
essentially is compressing the movies by only storing one copy of any
particular cast member).

The disadvantage of this animation method is that Director has to
load everything into memory before it can start playing and, there-
fore, consumes much more memory. Director needs to store all the
cast members in memory, whereas QuickTime needs to store only a
couple of frames in memory. Further, if you use mattes (which enable
one object to appear through gaps in another object), generating a
frame requires much more processing. This increase in processing
results in a much slower playback than that of a movie that has been
precomputed into one frame.

Figure 14.27 illustrates the different methods QuickTime and Director
use to store animation.

The disadvantages of Director are most apparent if you are creating
a large project that involves the linking together of several separate
animations. The pause during the loading of the next movie can
be excruciatingly long, particularly if the movie is playing from a
CD-ROM.

That's not to say that Director doesn't have it's advantages.
QuickTime does not currently support interactivity (buttons that can
be clicked, scripts, and so forth), and you can't create animation in
QuickTime—you need another tool. Further, Director can export
animations to QuickTime.
What is it good for?

Director can be used for just about anything. The animation capabilities make it particularly attractive for projects that require a lot of visual action. Director is not particularly strong with text, and if you are developing what is primarily a button-clicking interface with little
or no animation, then you may be much better off using HyperCard, SuperCard, or another available tool.

Although QuickTime enables applications such as HyperCard to play animation, you must use another tool to create the animation. Director is a good tool for that job.

Director supports a feature similar to XCMDs, called XObjects. XObjects provide a capability in Director similar to that provided by XCMDs in HyperCard, but with a slightly different format. In some, but not all, cases, Director can work with some XCMDs, by using a special XObject that knows how to communicate with XCMDs. Director also has a sophisticated scripting function, called factories, that enables you to create very complex routines and functions. Beginners should avoid using this function, however, until they are completely comfortable with Director’s Lingo language.

**Where to go from here?**

Macromedia and certified trainers around the country offer courses and classes on using Director. If you are a beginning Director user, and need to get something done quickly, you may benefit from attending one of these courses. Director seems to be one of those programs that lots of people have—or think they should have—but that few people know how to use. It can take a while to grasp the concepts of how Director works, and that’s why a hands-on class could make the difference between a successful project and a product box gathering dust on the shelf.

*Using MacroMind Director, Tony Bove and Cheryl Rhodes. Carmel In, Que 1990.* A little long in the tooth, but has the basic concepts and is about the only book available.
Other programs to consider

Some other programs are currently available that you may want to consider using for the development of your multimedia project. The most obvious of these is another Macromedia program, Authorware. Authorware was originally designed for producing training programs and is very good for sequential presentations and question-and-answer-format presentations. Macromedia has updated Authorware, and you can now play Director animations directly within Authorware presentations. Authorware is also cross-platform—it runs on Macintosh and Windows—which is an important consideration for some projects. The more formal structure of Authorware (you create the production's structure in a linked list that resembles the logic diagram for a computer program) is an easy way to enter some forms of productions such as training programs.

You also can choose from among some iconic programming languages for the Macintosh. Two such environments are Serious and Prograph. Although you still have to learn a programming language to use them, these tools provide some basic building blocks for creating your own programs, as well as a different interface for designing the logic of a program using icons and links between these icons. Although some people consider iconic programming languages to be easier to use than traditional programming environments, they are more complicated than programs such as Director or HyperCard. The iconic programming languages, however, also provide much more power.

In choosing an environment, you must consider a number of questions: Which tool will best suit your needs? How long will it take you to learn the tool? And how long will it take to produce the final production? Sometimes you must trade some functionality in the final production for a reduction of time spent in learning and using more sophisticated tools. Only your individual needs can determine your decision.
This chapter has reviewed the most sophisticated multimedia authoring tools available. The next chapter looks at some of the interactive presentation packages currently on the market.
Chapter 15

Interactive Presentations
The presentation programs covered in this chapter, Action! and Special Delivery, represent a midpoint between the more traditional slide-based presentation programs PowerPoint and Persuasion, and the authoring environments Director, HyperCard, and SuperCard. The presentation programs add buttons and links to the basic model of a sequential slide presentation, to enable you to create interactive presentations. These programs, however, don't provide all the features and the scripting capabilities of the authoring environments, and so are covered separately in this chapter.

The information in this chapter tells you what presentation programs are, what they do, and how you can use them in your work. The chapter discusses both the capabilities and the limitations of working with presentation programs. The final sections of this chapter provide details about two popular presentation programs, and offer some predictions for future developments to watch for in this newly emerging field of multimedia environments.

Understanding presentation programs

Although you cannot use presentation programs to create the sophisticated interfaces or complex programs possible with authoring environments, they are perfectly acceptable for creating interactive sales brochures, most kiosk applications, and demos. You also can use presentation programs to create training and help systems.

Currently, you cannot add functionality to presentation programs by using plug-in modules, such as the XCMDs used with HyperCard and similar programs. Future presentation programs, however, may support the addition of plug-in modules.

The category of presentation programs includes Macromedia Action!, Interactive Media Corporation's Special Delivery, and Astound! from Gold Disk. MovieWorks and Cinemation also fall into this category because they offer buttons and interactivity, but these programs come from an animation—rather than presentation—background and therefore don't resemble traditional presentation programs. Chapter 8, "Animation," describes the MovieWorks and Cinemation programs.
Using interactive presentations

Presentation programs are marketed primarily to users who create sequential presentations and want to add the interest of interactivity to their presentations. As mentioned earlier, presentation programs are good tools for creating such presentations as kiosk applications, interactive brochures, training systems, and demos.

Adding interactivity, however, also adds a level of complexity that you may not want in your presentation. Do you really want to be trying to remember which button to press while nervously making a presentation to the president of the company? And when making a presentation, how often do you really need to branch off in different directions for different audiences? You may want to start and stop a QuickTime movie or an animation at an appropriate time, but that may be the extent of interaction you need in the presentation.

Another problem involved in creating animations and graphical presentations is that doing so requires a skill many presenters lack. Action! goes a long way toward solving this problem by providing a large collection of document templates that users can adapt for their own presentations.

In spite of these problems, presentation programs do have application in the areas previously mentioned, and as a multimedia producer you may want to have one of these programs available for simple presentations. Action! and Astound! both can create standalone documents that you can distribute free of charge.

Using Action! and Special Delivery

The remainder of this chapter is devoted to a brief description of the two programs Action! and Special Delivery. These two programs provide a good overview of the capabilities of presentation programs.
Although the programs offer similar features, they have very different interfaces.

**Action!**

Action!, from Macromedia, is available in Macintosh and Windows versions. Although the interface and functions are quite similar in both versions, the current programs do not share a common file format. As a result, you cannot move a presentation created in Action! on the Macintosh over to the Windows platform (Macromedia is working on offering that functionality). Figure 15.1 shows a presentation being edited in Action!.

![Figure 15.1. Editing a presentation in Action!](image)

You organize presentations in Action! into **scenes**. Scenes essentially are slides with any animation that occurs while that slide is visible. Scenes can be viewed in the Scene Sorter window, which provides thumbnails of the individual scenes (see figure 15.2).

Each scene in an Action presentation begins at a specific time within the presentation; each scene has a specific duration (for example, two minutes). You also must define the time at which an object appears in the scene and how long the object stays onscreen. You
make these definitions with the Edit Object dialog box (shown in figure 15.3) or in the Time window (at the bottom of the screen shown in figure 15.2).

**Figure 15.2.** The Scene Sorter and Time window.

You can interrupt the presentation by adding a pause at any time within the Time window. The Time window displays the start time and duration of all objects in a scene, and can be used to alter these parameters by clicking and dragging.

**Figure 15.3.** The Edit Object dialog box.
You can turn any onscreen element into a button by defining a Link parameter in the Edit Object dialog box. **Linking** adds interactive elements to a presentation such as going to another scene or returning to the last scene.

While creating time-based presentations is much more complicated than creating regular slide-based presentations, Action! makes it easier for those starting out by providing a collection of templates that you can use to create a new presentation quickly. You also can create your own templates.

### Special Delivery

Special Delivery differs greatly from Action! (and most other presentation programs) in the way you define interactivity within the program. The basic building blocks of a Special Delivery presentation are slides, similar to those in Action!; but you define interactivity in Special Delivery by drawing links from buttons to the screen or other onscreen elements. Figure 15.4 shows the creation of a presentation in Special Delivery. At the bottom of the screen in the figure, you see the Map section, which provides thumbnail views of individual screens from the presentation.

**Figure 15.4.** A presentation in Special Delivery.
To place a graphic, a piece of text, or a QuickTime movie on the screen, you first draw a portal. A portal is a rectangle that "holds" the item. You can use portals to crop and resize graphics, as well as to add interactivity. Special Delivery has three kinds of portals; rectangle, rounded rectangle, and oval.

A separate layer in the slide contains the interactive links. You create these links by selecting the button tool and dragging from one portal to another portal, or to the screen. A predefined collection of commands is used to perform different functions, such as playing QuickTime movies or moving to other screens. Special Delivery supports several basic transitions for moving from one scene to another. Figure 15.5 shows a link being defined in Special Delivery. When the button being defined in this figure is pressed during a presentation, a QuickTime movie plays. Figure 15.6 shows how to create a link from a button to an object—for example, between a button that starts a movie playing and the movie itself.

![Diagram of portal link](image)

**Figure 15.5.** Defining a Link in Special Delivery by dragging from one portal to another.
Like Persuasion and PowerPoint, Special Delivery supports speaker notes (a printed document for the speaker or attendees that shows a reduced image of the screen and some text notes). Special Delivery also can display these notes on one monitor and the presentation on a second monitor, if you use a computer that is connected to two monitors.

**The future**

The three presentation programs mentioned in this chapter (Action!, Special Delivery, and Astound!) appeared in less than one year—amazing, when you consider that prior to their arrival, no other program existed in this category. What does this quick development mean for the coming years? If these programs are popular, their manufacturers undoubtedly will update the programs with more features. The most likely updates are support for scripting, some kind of plug-in functionality (XCMDs), and support for AppleEvents. With such updates, these programs will rival the capabilities of the programs described in Chapter 14, "Authoring environments."

The next chapter goes step-by-step through the process of creating a prototype using HyperCard. For those just starting out, this chapter will serve as an introduction and tutorial for creating multimedia projects.
part V
The multimedia workshop
Putting it to work
You've looked at the design examples, and read about all the tools and authoring environments. Now fasten your seat belts and get ready to put it all to use! This chapter takes you through creating a project using HyperCard. Starting at the beginning, we will assemble an electronic interactive publication—the project that was specified in Chapter 3, "Designing a multimedia project."

What is the project?

*Exploring Multimedia* is a monthly Macintosh multimedia newsletter whose primary audience is Macintosh multimedia producers and users. There's only one problem—it appears on paper, and while a paper version is the easiest to produce and distribute, the editor would like to have an electronic version that can be distributed through bulletin boards and on disk as a promotional item.

Using material from the first issue of the newsletter (the text and pictures), the electronic version will be assembled in HyperCard. This is a reasonably easy exercise—all that's required is a main card that contains the table of contents (the articles in the issue) and then a series of cards for each article. For this beta^1^ version only one article will be implemented.

The complete publication will look something like figure 16.1, but for this exercise we will implement only the main card and one article.

Getting started

Included on the disc, in the folder Workshop files, is a collection of data files that make up the first article (which is about the release of QuickTime 1.5). The images are in 8-bit color—though you should be able to use them on a grayscale monitor. Included is the text for the article, as well as three HyperCard stacks. One of the stacks (Issue No. 1) will be used to build the prototype. Another stack (Table stack) contains a table used on one of the pages. The last stack (Colorizing HyperCard 1.0) contains the XCMDs used to display color images in HyperCard. This stack is primarily included for documenta-
tion purposes. It explains and demonstrates how the routines work, as well as many of the commands and options available that are not used in the prototype. You might want to keep this stack for future reference. There is also a file called Chapter 16 scripts, which contains most of the scripts mentioned in this chapter. You can copy and paste the scripts from this file into HyperCard.

**Figure 16.1.** The operational diagram for the electronic version of Exploring Multimedia.

You should copy the folder Workshop files to your hard disk before you begin work. It's important that the graphics are in the same folder as the stack that you create. If they aren't, the routine that displays them won't be able to find the graphics.

The ColorizeHC XCMD has already been copied into the Issue No. 1 stack, so you don't need to install it yourself. If you are running HyperCard under MultiFinder on System 6, or you are using System 7, you will have to increase the amount of memory that HyperCard is using to at least 2 megabytes. You must do this because the ColorizeHC XCMD uses memory. If you don't increase the memory available, the routine will not work and you won't see the color graphics.

To increase the memory allocated to HyperCard, while HyperCard is not running, select HyperCard (by clicking it once) and then press Command-I to display the Info window. Change the Current Size in

---

2To install the XCMD you would need to use either ResEdit or some other resource editor.
3Note: This assumes that you have available a copy of HyperCard that you can use to edit HyperCard stacks. Depending upon when you purchased your Macintosh, you might have received a “full” copy of HyperCard, a locked version, or the Player version (which cannot be used to edit stacks.) If you have the locked version of HyperCard, you can unlock it by going to the last card in the Home stack (by pressing Command-H); press Command-4 (to display the User Level). If the card shows only two levels (1-Browsing, 2-Typing,) press Command-M (to bring up the message window) and type Magic, then press the Return key. The other 3 levels (3 through 5) appear. Click the Level 5-Scripting button to start working.

You can also buy the HyperCard developer kit from dealers and mail-order companies.

the Memory section of the Info window to at least 2,000K, or 3,000K if you have at least 5M of memory in your computer.

Step 1: Setting up the opening screen

Open the stack called Issue No 1.3 The stack contains only one card; it is blank. The first step is to make this card solid black (it’s what the user will see when the stack starts up.) Choose the paint bucket tool, choose solid color from the Pattern palette, and click in the middle of the card. The card should now be black.

Now add another card. To do this, choose New Card from the Edit menu (Command-N). A white card will now be visible. To go to the previous card (the first one you created), choose Prev from the Go menu. HyperCard displays the black card—the first card you created.

When the stack is opened, we want HyperCard to automatically go to the second card. To do this, choose Card Info... from the objects menu (see figure 16.2), and click the script button. Type the following script in the script window (see figure 16.3):

```plaintext
on OpenCard
    lock screen
    go to next card
end OpenCard
```

Click on the close button (yes, you do want to save the script). Now test the script. Go to the second card (from the Go menu choose “Next”) then go back to the first card, by choosing Prev again.

What happens? The screen should briefly go black and then white again as you go to the first card, and then HyperCard automatically transfers to the second card.

What does this accomplish? Not much. Except that it shows how to get HyperCard to do something when a card is opened, as well as how to go to the next card in a stack. The real reason this card and the instructions were added was because of the function lock screen.
When HyperCard locks the screen, it can go to any card and change what is on the screen, and you won’t see anything change until the script finishes executing, or the command `unlock screen` is executed.

![Card Info dialog box](image1)

**Figure 16.2.** The Card Info... dialog box.

![Script window](image2)

**Figure 16.3.** The script window.

This is important because we are going to use the ColorizeHC routine to add a color background to the cards in this stack. The problem is that it takes a few seconds for ColorizeHC to display the background. If we just went to the card and then displayed the color background, the user would see the buttons and fields we had placed on the card for a fraction of a second before the color background appeared. This wouldn’t look very good.

By displaying a blank, black card first, then locking the screen while HyperCard gets its act together and displays the second card, we can provide the best looking “experience” for the user. The same technique is used when going from card to card in the rest of the publication.
Displaying color graphics

Now you will add a color background to the first card. This involves adding to the second card an OpenCard script that uses the ColorizeHC routines to display the PICT file Master Bkgnd. The file was created in Photoshop, and is included on the disc.

Open the script window for the second card and add the following script (SCRIPT ONE):"'

```plaintext
on openCard
  global pathname
  lock screen
  put pathname & "Master Bkgnd" into the_picture
  colorizeHC "new", the_picture, "0,0,512,384"
  if the result is not empty then answer the result
  unlock screen
end openCard
```

Now close (and save) the script, and try going back to the first card (which takes you automatically back to the second card). If you did it right, then you should see the following error message (see figure 16.4).

![Error loading PICT resource dialog.](image)

**Figure 16.4.** Error loading PICT resource dialog.

Don't worry (only worry if you didn't get that message!) The ColorizeHC routine needs to know where on your hard disk the file is. That means providing the pathname (which looks like Diskname:foldername:foldername:) Luckily, we can have HyperCard provide that for us by writing a short routine which we'll call GetPath. The script appears below.

We could call the routine every time we wanted to display a PICT file, but that would be a waste of effort. Instead we'll call the routine once when the stack is opened, and save the pathname in a global. A **global** is a piece of memory in which a value can be stored and which remains intact as long as the stack is running (as opposed to local variables, which only last as long as the particular script is running).
Open the Stack Info... dialog, and add the following script to your stack (SCRIPT TWO).

```plaintext
on OpenStack
  getPath
end OpenStack

on GetPath
  global pathName
  put the long name of this stack into LongPath
  put the short name of this stack into ShortPath
  put the number of characters in LongPath into longLength
  put the number of characters in ShortPath into shortLength
  put char 8 to (longLength - (shortLength + 1)) of LongPath into pathName
end GetPath
```

This is actually two scripts. The first one, OpenStack, is called when the stack is opened, and calls the second script called GetPath. The script GetPath stores the pathName in the global pathName, which is already in the original script (we just hadn’t set the value up yet!)

Now close (and save) the script, exit the stack by choosing Quit, and then start the stack again. This time you should see the color graphic on your screen, just like figure 16.5 (but with color).

![EXPLORING Multimedia](image)

**Figure 16.5.** The color graphic as it appears on the card.

Notice that the background has buttons drawn on it; these buttons don’t work yet.
Adding a Quit button

Choose the button tool from the Tools palette. Hold down the Command key and the cursor will change to a cross-hair. Drag out a button that covers the Quit button (see figure 16.6).

![Figure 16.6. Dragging out a button.](image)

With the button tool still selected, double-click the button. This will open up the Button Info... dialog box. (Double-clicking is a short cut for clicking once on the button, to select it, and then going to the Objects menu and choosing Button Info...) Check the Auto Hilite check box (figure 16.7).

![Figure 16.7. The Auto Hilite check box.](image)

Click the Script... button and add the following script to the button (SCRIPT THREE):

```script
on mouseUp
    domenu "Quit HyperCard"
end mouseUp
```

This script executes a command that is in one of HyperCard's menus—the Quit HyperCard command. You can use the domenu
instruction to execute any command that is available in HyperCard's menus from a HyperCard script.

Close the script window, and choose the browse tool (the hand in the Tools palette.) Clicking on the Quit button should now quit HyperCard.

Step 2: Adding another background

The next step is to add a new background in the stack. This background will be used to display the content cards—the cards that contain the text and pictures that make up an article.

Each article will consist of a sequence of cards that share this common background. The background will contain a field for storing the text, as well as buttons for navigation, and standard routines for displaying the color background and any picture elements.

Add a second background

Open the stack (if it isn't already open). You should be on the second card (with the color background).

From the Objects menu, choose “New Background”. The screen will change—the menu bar will have a dashed pattern surrounding it. This indicates that you are editing the background. A new card has also been added to the stack. This won’t be immediately obvious, because the same color background is visible.5

First we will add a command that will change the color background—it's too confusing to still see the color background from the first card!

From the Objects menu choose the Bkgnd Info... command, and then add the following script (SCRIPT FOUR):

```
on openCard
  global pathname
  lock screen
```

5I'm confused—we created a new background but the color background is still the same! What's going on?

The color backgrounds displayed by the ColorizeHC XCMD are completely separate from the Backgrounds that HyperCard uses. The ColorizeHC XCMD is like a background to the Background of HyperCard, and it stays on screen until it is changed—usually by sending another command like “New” or “Dispose” to the XCMD. Until that happens, the color background will remain the same, no matter which HyperCard background we are in.

Just remember that the HyperCard background is used to store buttons and fields that are common to several cards.
put pathname "Article Background" into the_picture
colorizeHC "new", the_picture,"0,0,512,384"
if the result is not empty then answer the result
unlock screen
end openCard

Close and save the script. This script is exactly the same as the script in the first card, except this script is in the background (so it will be executed every time a card in this background is opened) and it displays a different file—Article Background.

From the Edit menu choose Background (or press Command-B). This should switch you out of Background editing mode (the pattern around the menu will disappear, and the item Background in the Edit menu will be unchecked.)

Go to the previous card (Command-2) then go to the new card (Command-3). You should now see a different color background (see figure 16.8).

Figure 16.8. The color background for the article cards.

Add a background text field and buttons

Now we will add a background field and a background button, and start placing the text into the card.
Choose Background from the Edit menu (or press Command-B) to edit the contents of the background. Choose the Field editing tool from the Tools palette. Hold down the Command key and drag out a text field to fill the area between the *Exploring Multimedia* logo and the buttons at the bottom of the screen (see figure 16.9).

![Dragging out the field](dragging_out_the_field)

**Figure 16.9.** Adding a background field to contain text.

Now change the default font used to display the text. To do that, double-click the text field (with the field tool still selected). When the Field Info... dialog box appears, click on the Font... button, and then choose the font you want to use in the field (see figure 16.10).

![Choosing the field's default font](choosing_field_font)

**Figure 16.10.** Choosing the field's default font in the font dialog box.

In this case I am using 10 point Geneva, because I know that everyone will have that font, but you can use any other font and size that you like.
Close the font dialog box by clicking the Okay button.

Now add a button to the background. Choose the button tool, then hold down the Command key and drag to create a button over the "Contents" area on the background.

Double-click the button (to bring up the Button Info... dialog box) and click the Link To... button. The Link To palette appears. Now go to the previous card by pressing Command-2 and then click on the This Card button. HyperCard will now jump back to the original card.

Open the script for the button (a shortcut is to hold down the Option and Command keys and click on the button). You will see a script that resembles this one:

```
on mouseUp
    go to card id 3593
end mouseUp
```

Don't be surprised if the ID number is different—HyperCard randomly assigns ID numbers to objects. As a finishing touch, add the line lock screen in front of the go to card line, so that the script looks like this:

```
on mouseUp
    lock screen
    go to card id 3593
end mouseUp
```

Don't change the card ID number—leave it as the number that you already have.

Now add the Previous and Next buttons in the same way—except don't use the Link To command; instead, just add the following scripts:

**PREV button (SCRIPT FIVE):**

```
on mouseUp
    lock screen
    go to previous card
end mouseUp
```
NEXT button (SCRIPT SIX):

```javascript
on mouseUp
  lock screen
  go to next card
end mouseUp
```

These two scripts move from one card to the Next card, or to the Previous card.

Now it's time to start adding some content.

## Step 3: Adding content

The text of the story is contained in the document “QuickTime 1.5 text.” The illustrations for the article are in the following PICT files:

- Movie size
- Dithering No1
- Dithering No2
- Text track
- Comp settings

The text and images will be displayed on a series of the article cards.

You should be on the first content card of the stack. If you are currently editing the Background (the menu has the dashed lines around it), then choose Background from the Edit menu to switch back to the card.

### Add a title

First, we’re going to add a title. We are going to create this in a large display typeface. Since the user might not have this typeface, it will be created as a bitmap (painting) on the card.

Choose the Text tool from the paint tools in the Tool Palette. Click next to the logo and add the text “QuickTime 1.5” (see figure 16.11).
To change the size and font, press Command-T; the font dialog box appears. Choose any font and size you like, but choose something reasonably large—say 30 or 40 point.

Figure 16.11. Adding the title text.

Click somewhere else on the card to "set" the text—do this instead of pressing the return key to add another line. (If you simply press return to add a second line, the new line will have the same font size as the first.) Click underneath the heading and add a second line that reads "Photo CD and a new compressor," in a smaller size of the same font.

If you don't like the placement of the text, you can move it (see figure 16.12). Use the marquee or lasso tool to select the piece of text, and drag it to another location on the screen.

Figure 16.12. Moving bitmapped title.
Copying text

Open the “QuickTime 1.5 text” file. Copying the text from your word processor to the HyperCard stack is a slightly tedious process—you have to copy some text, paste it into the card, then add more text until the field is full.\textsuperscript{6}

Since you don’t want to copy too much text, set up the font and size of the text in your word processor to be the same as on the card. This will give you a very good idea of how much text to copy (see figure 16.13). (The process will be much easier if you can run HyperCard and your word processor at the same time.)

\begin{figure}
    \centering
    \includegraphics[width=\textwidth]{figure16_13.png}
    \caption{Selecting text to copy into the HyperCard stack.}
\end{figure}

Go to the card and with the browse tool selected, click in the text field (remember, the text field we created in the background of the article card.) Paste the text into the card. You should now have a card with text, much like in figure 16.14.

If you didn’t paste enough text, or pasted too much, you will have to either select and copy more text, or remove some text (being careful to save the text for the next card).

In figure 16.14, too much text was selected, so the last line was removed (see figure 16.15).
QuickTime 1.5

Photo CD and a new compressor

Apple has upgraded QuickTime with version 1.5, adding several new features and improving performance. While you can make use of some of these features immediately, others require developers to update their products.

We've had QuickTime 1.5 for about a month now—long enough to play with, but not long enough to fully explore or fully appreciate the changes. Over the next few months we'll be referencing to QuickTime to see how these new features are being incorporated into every application.

Compact Video
QuickTime 1.5 adds a new compressor/decompressor (codec) which compresses more than the Video compressor. Called the Compact Video compressor, this new codec is able to double the size of movies on existing hardware. There are an LC and an LC+ video 120 movie at 12 frames per second with the Video Compressor, using the Codec, the same LC+ is able to play a 240 x 160 movie at 12 frames per second. The savings (see Table 1) mean that these movies play well from CD-ROM. Just the Video compressor can play these highly compressed movies at high frame rates. The incompressible of similar compression savings, the LC+ takes to decompress is for use in movie playback.

An important feature for movie producers is that the Compact Video

Figure 16.14. Pasting text into the HyperCard article card.

QuickTime 1.5

Photo CD and a new compressor

Apple has upgraded QuickTime with version 1.5, adding several new features and improving performance. While you can make use of some of these features immediately, others require developers to update their products.

We've had QuickTime 1.5 for about a month now—long enough to play with, but not long enough to fully explore or fully appreciate the changes. Over the next few months we'll be referencing to QuickTime to see how these new features are being incorporated into every application.

Compact Video
QuickTime 1.5 adds a new compressor/decompressor (codec) which compresses more than the Video compressor. Called the Compact Video compressor, this new codec is able to double the size of movies on existing hardware. There are an LC and an LC+ video 120 movie at 12 frames per second with the Video Compressor, using the Codec, the same LC+ is able to play a 240 x 160 movie at 12 frames per second. The savings (see Table 1) mean that these movies play well from CD-ROM. Just the Video compressor can play these highly compressed movies at high frame rates. The incompressible of similar compression savings, the LC+ takes to decompress is for use in movie playback.

An important feature for movie producers is that the Compact Video

Figure 16.15. Selecting excess lines and deleting them.

Formatting text

Now that the text is in place, it would be nice to add some formatting. HyperCard does not support tabs, and no text attributes came across (bold, italic, and so forth). We can imitate tabs with spaces, and HyperCard does support styles and other typefaces within a text field.
To imitate tabs at the beginning of each paragraph, simply go to the beginning of a paragraph and add some spaces (three or four will do—just make sure you are consistent).

To add formatting to the subtopic “Compact Video”, select the text and press command-T. The text dialog box will appear. Choose bold, and close the dialog. The card should now look something like figure 16.16.

**Figure 16.16. Formatted text.**

### Step 4: Adding a picture to a card

The next card has a single picture on it with a caption. The appearance of the final card is displayed in figure 16.17.

The ColorizeHC routine will also be used to display this graphic (which is a PICT file.) By using the “Add” option instead of the “New” option when we call ColorizeHC to display the graphic, we can tell ColorizeHC to add this graphic to the existing color background, rather than replace the background. We will also need to tell ColorizeHC where to display the graphic on the card.

We could simply replace the standard OpenCard routine in the background with a routine for each card that displays the
background and adds a graphic. But this would require a lot of duplication—and all we want to change from card to card is the name of the graphic and the location.

There's a way to handle this problem that will make it much easier to add graphics to a card.

![Image of an article card with a picture displayed in it.](image)

**Figure 16.17.** Article card with a picture displayed in it.

### Adding a generalized graphic display

Rather than change each script, we are going to add a background field that will contain the name of the background image to be displayed, as well as the name of the graphic image we want to display on the card. We will also place a button on the card which will be used to indicate the location for the graphic.

The background routine will be changed so that on OpenCard, the routine will look in the background field we added to find the name of the background image to display. If there is a second file name in the field, the routine will find the location of the button and display the graphic at that location.

Here's how to do it:

From the Edit menu choose Background, then add a new field on the card (by selecting the field tool, holding down the Command key, and
clicking and dragging out a field). The field will be hidden in the final version, so you can place it anywhere, but a good place is at top right of the card.

Name the field. To do that, double-click the field, and in the field name type: PICT Info (see figure 16.18).

![Figure 16.18. Naming the new field.](image)

Switch back to the card (Command-B), switch to the browse tool, and in the first line of the field type: Article Background.

Now edit the Background script. Replace the script with the following (Script Seven):

```on OpenCard
    global pathname
    lock screen

    put pathname & line 1 of background field "PICT info" into the_picture
    colorizeHC "new", the_picture,"0,0,512,384"
    if the result is not empty then answer the result
    put the number of lines in background field "PICT info" into x
    if x > 1
        then
```
repeat with y = 2 to x
put "Add" & (y - 1) into button_name
put the topleft of card button button_name into illus_location
put pathname & line y of background field "PICT info" into the_picture
colorizeHC "add", the_picture, illus_location
end repeat
end if
end repeat

unlock screen
end OpenCard

Close and save the script. Notice that this script gets the background name from the first line of the new field we added. If there is more than one line in that field, the routine finds the location of a card button called “Addx” (where x is a number starting at 1 and going up in sequence). This location is then used to add a graphic to the background image, the name of the graphic coming from the field.

This routine will add several graphics if more than one line is added to the background field.

Before adding the second card, check to see that this routine works. Go to the first card and then switch back to the first article card. The backgrounds should still switch between the two cards.

Now add a new card by pressing Command-N. You will get the Error loading PICT message (because there is no PICT name in the background field.) Using the Browse tool, position the pointer over the text field, and simply type Article Background in the background field (see figure 16.19).

Now add a button by choosing the button tool, holding down the command key, and dragging out a button just under the Exploring Multimedia logo. Name the button by double-clicking it and in the button info dialog typing Add1. Now select the browse tool, and add the words Movie size (the name of the graphic file you want to add) in the second line, underneath the “article background” name.

Go to the previous card, then go back to this new card. You should now see something resembling figure 16.20.
What if it doesn't work?

If it doesn't work, it could be because of an error in the script (did you copy the scripts exactly?) or because you didn't name the button, the field, or the images correctly. Note that the button must have been created on the card, not on the background (a card button is different from a button on the background).

Also, you must put the background name in the first line of the field, and the graphic in the second line. This can be hard to do when the shape of the field isn't visible. To help you, go into the field info dialog box for the field and click the Show Lines and Rectangle options. It will make setting the field up much easier.
Finish up the card

To change the position of the graphic, simply select the button tool, and click and drag the button to a new location. To finish up the card, add a caption. You can do this simply by clicking at the bottom of the card (in the background text field that's on the card) and adding the text:

The Compact Video Compressor doubles the size of movies playable on CD-ROM from 180 x 160 to 240 x 180.

Step 5: Adding the other cards

The third card has only text on it (see figure 16.21). You can, of course, change the layout if you like, but that's the way the original prototype was set up. Create a new card (Command-N), add Article Background to the PICT Info field (so that the background is used), and add the text. Remember to format it with spaces for tabs.

Figure 16.21. The third article card.

Remember that as you add new cards you must add the text Article Background to the field PICT Info so that the color background is displayed.
Manually wrapping text

The fourth article card (figure 16.22) has a table. On the prototype this was created as bit-mapped text. When creating tables it is sometimes easier to work with bit-mapped text than to try to rearrange and align fields.

![Table Stacks](image)

**Figure 16.22.** The fourth card has a table, and the text has to be wrapped.

You can recreate the table, or you can copy it—there’s a copy in the stack Table Stack.

One problem with placing text into this card is that the text will overlap the table. There are two solutions to this problem. One is to create a unique card field which is sized correctly. This is probably the easiest thing to do—and you are welcome to do that if you want. Alternatively, you can manually rewrap the lines by placing a return at the end of each line so that the lines aren’t too long (see figure 16.23).

I used the latter method because I wanted to create an automatic index of the stack, and this was easier to do by keeping everything in one field. It makes searching easier if you only have to look in one field—though you could probably use a method similar to the one that we used with the graphics: have a background field that contains the name of a card field if a card field is used to place the text on the screen.
Cards 5 through 10

Finish the article adding the rest of the text, and the other graphics as necessary (as in figures 16.24 through 16.29).

Figure 16.24. The fifth card contains graphic Dithering No1.
Putting it to work

Figure 16.25. The sixth card contains graphic Dithering No2.

Figure 16.26. The seventh card contains text only.

Figure 16.27. The eighth card has a picture and text.
Figure 16.28. The ninth card has a picture and text.

Figure 16.29. The tenth card is text only.

Step 6: Some finishing touches

You may have noticed that in the “final” version there is a small subheading on each card listing the title for the article. This was added as a paint text element. Also, between the Next and Previous button there is a field that contains the number of the card, and the total number of cards (1 of 10, 2 of 10, and so on).
Add the field as a background field, and then manually type in the information. (You should center the text to make it look good.)

You should lock the text fields (go into the Field Info... dialog box for each field and click the Lock Text check box—see figure 16.30).

![Figure 16.30. Locking the text field.](image)

**Hiding the background field**

The background field containing the list of graphics needs to be hidden. We can do that easily by bringing up the message window and executing a HyperTalk command in the window. Press Command-M to see the Message window. (You can use this window to execute one line HyperTalk scripts.) Type the following line:

```
hide background field "PICT info"
```

Then press the return key (see figure 17.31). The Background field should now disappear.

**Adding a button to the first card**

We need to add a button that links the first card of the stack (the table of contents) to the article. Go to the first card, and choose the button tool. Hold down the command key and drag out a button
below the logo. Type the title for the button using the text paint tool. If you want to make the button visible, go into the Button Info... dialog and click Rectangle. Then link the button to the first card of the article.

![Image of QuickTime 1.5](image)

Figure 16.31. Hiding a background field.

**Things that haven’t been done**

This project could be expanded by adding sound and QuickTime movies. HyperCard supports sound (you have to place sound resources in the stack and then use the command `Play (sound name)` to play a sound). There are XCMDs for playing QuickTime movies that are available from Claris. You can even add animation using the XCMDs for the programs Director, ADDmotion, and Animation Works. Be warned that as you add these extra functions, the stacks will become larger and more complicated and you may have to increase the memory allocation for HyperCard again!
As a finishing touch, we have yet to add an About... screen (which explains what this stack is), or even a Help screen(s). If you want to add these, you should probably create another background with a different background picture, and then add buttons to the first screen that link to these cards.

You could also add another article after the first one. One interesting problem is the Prev-Next buttons. Should they take you to the next article, or should the Prev button not be available on the first card, and the Next button not available on the last card of an article? That’s for you to decide. In this prototype, clicking the Next button on the last card takes you to the next card, which is the first card in the stack, so you end up at the table of contents. Similarly, the Prev button on the first card in the article also takes you to the table of contents card.

If you do decide to change the operation of these buttons so that they only take you to the next or previous card in the article (you would use the table of contents button to exit from the article), you will have to hide the next and previous buttons in the background on the first and last card of the article. That would mean either creating two new backgrounds (without the appropriate buttons) or you could take a small area of plain color as a graphic and display that on top of the background, or you could take the buttons off the background and display the buttons as separate elements. You will also have to alter the scripts of the buttons.

Performance-wise, the first option would probably be the best and easiest to do, while the second option is the most efficient—you’d only need one graphic and only have to display it on the first and last cards.

Other functionality that you might consider adding is an index or dictionary of words used in the articles. In the Final Example folder you will find that a Dictionary stack has been added (see figure 17.32). This stack was generated from the original article, and lists the occurrences of certain words throughout the article. When you click on a word, the different locations are shown, and clicking on one of these displays the context of the word. Clicking on the context description takes you to that article.
The Dictionary is not complete—it does not hold definitions and has a few minor problems (like showing the card ID instead of the article card number) but you might want to pick the stack apart to see how it was created.

Moving on

Using what you have learned here, you can now start creating your own HyperCard stacks to hold other information. You should consider getting a HyperCard book to help you learn scripting if you want to create more complex projects.

There are a lot of projects that have been created in HyperCard and other tools, and you should try and see as many as possible to get ideas for implementing your own projects.

Finally, don’t be afraid to plunge in and do something. As you have seen in this chapter, it is very easy to put together a simple project using a tool such as HyperCard. As long as you start with a clear design, work in chunks, and experiment as you go, you should be able to create your own projects with too much trouble. Good luck!
What's on the disc

MACINTOSH Multimedia Workshop

Includes:
Multimedia illustrations for Macintosh Multimedia Workshop
Demonstration programs
QuickTime extension
Utilities
Sample multimedia projects
Exploring Multimedia bulletin board interface
Exploring Multimedia newsletter and subscription offer

© 1992 Hayden Books and its licensors
All rights reserved
To order more Hayden books, call
1-800-428-5331

Michael D. Murie Hayden Books
The CD-ROM included with this book includes a wide range of things—from demos and sample file to images and QuickTime movies.

Different applications and files on the disk have different requirements; we recommend a color Macintosh with System 7 and 4M of memory (and a CD-ROM drive!) as a minimum. Some of the applications require more memory, and some must be copied to your hard disk to work. The QuickTime movies must be viewed with an application that can play QuickTime movies (CameraMan's MoviePlay and Adobe Premiere can do so.) You must also install the QuickTime extension, which is included on the disk. Some demos (and the "Putting it to work" files) require HyperCard.

Some of the applications on this disc are shareware. These applications are provided to you on a trial basis. If you like them, you should send the appropriate shareware fee to the author of the application. (Read the Read Me files with each application to find out more.)

The purchase of the *Macintosh Multimedia Workshop* does not include purchase of these programs. Please support shareware authors.

The following is a short description of the contents of the disc:

**Tour the disc**

This is a QuickTime movie that is a guide to the contents of the disc. The movie was captured using CameraMan (see Chapter 10, "QuickTime") and the narration was added using the CameraMan editor. Note that MoviePlay (a utility for playing QuickTime movies that is distributed with CameraMan) is included on the disc and can be used to play this and other QuickTime movies.

**QuickTime**

This is version 1.6 of the QuickTime extension. Drag it to your System Folder to install it (you'll have to restart your computer, too). You
must be running a machine that supports QuickTime (see Chapter 10, "QuickTime," for more information).

Cover art

This is a PICT file of the cover of the book. Why is it here? Well, we had to put it somewhere!

Companies

This folder contains demo versions or sample files that were supplied to us by various multimedia software developers. The inclusion of a product in this folder does not constitute an endorsement from us. Rather, these are the companies that made material available to us within our schedule.

Vividus

Samples and self-running demo of Cinemation, the animation and presentation program. Copy the folder to your hard disk to run the demos.

Specular International

Demo version of Infini-D 2.0, as well as sample images, QuickTime movies, and an interactive presentation on Infini-D.

Playmation

Sample images and movies generated using Playmation, a 3D modeling tool. (Movies must be played with a QuickTime movie player.)
No Hands Software

Copy of Common Ground and demonstration documents. Common Ground is an electronic publishing tool.

Virtus

WalkThrough is a 3D modeling application. This folder contains sample images and movies created using WalkThrough. Also enclosed are three Voyager documents. These are 3D models with the playback software from WalkThrough included. They enable you to explore the 3D world, but you can't manipulate it.

Heizer Software

Demo versions of MasterScript, WindowScript, and CompileIt! These tools provide added functionality to HyperCard stacks.

Fractal Design

Demo version of Painter 2.0. Copy the folder to your hard disk to run the demos.

Multiple Media Tour

Catalog for the Multiple Media Tour, a multimedia clip art collection from Audio Visual Group.

Strata

Sample images generated using Strata 3D.
Adobe
Demo versions of Premiere 1.0, Photoshop 2.0, and Illustrator 3.0. Also included are some sample movies created using Premiere. Copy the folders to your hard disk to run the demos.

Movie Works
Demo version of MovieWorks, as well as sample presentations created using the program.

Morph
Demo version of Morph. Use this program to create magical transitions from one image to another.

PROmotion
Demo version of this animation package.

VideoFusion
Sample movies and an interactive presentation about VideoFusion. (Movies must be viewed with a QuickTime movie player.)

VideoToolKit
A demo movie created using VideoToolKit. (Movies must be viewed with a QuickTime movie player.)

CameraMan
MoviePlay, a utility for playing QuickTime movies. The "Tour the disc" movie was created using CameraMan.
This folder contains some multimedia productions that I have created.

**QuickTime Handbook movie**

This QuickTime movie was created as an advertisement for the book *QuickTime Handbook*, published by Hayden. The authors: David Drucker and Michael D. Murie. The movie was put together in a couple of days and was shown at an awards show.

The movie was created in Director and Infini-D (which was used to create the flying television.)

**Now & Zen**

Created almost four years ago as a parody of Apple’s Knowledge Navigator video tapes. This was created in Director.

**Farmers Market**

There are three presentations in this folder. The movies feature a selection of handcolored photographs by Susan Murie. The first presentation was created in Director, and has full-screen graphics. The other two are quarter-screen and are QuickTime movies. One movie is compressed using the Apple Video compressor, and the other using the Compact Video Compressor.

**Dogcow**

This movie was created using Director. There are two versions; one compressed using the Apple Video Compressor, and the other compressed with the Compact Video Compressor.
Exploring Multimedia

Exploring Multimedia is a monthly newsletter for Macintosh multimedia producers. On the disc there is an electronic copy of the first issue of the newsletter. There is also the software for calling the Exploring Multimedia BBS.

EM Vol 1 No 1 Grayscale

Electronic version of the first issue of Exploring Multimedia in Glue format (the viewer is enclosed). The images in this file are in grayscale.

EM Vol 1 No 1 No 2 B&W

Electronic version of the first issue of Exploring Multimedia in Glue format (the viewer is enclosed). The images in this file are in 1-bit black and white, so the file is much smaller, but the illustrations don't look very good.

Viewer

The application used to view and print the Glue-format documents.

Special Offer!!

A special offer to subscribe to Exploring Multimedia for $10 off the regular subscription price. Alternatively, call Multimedia Workshop at (617) 776-2469.

EM BBS Demo Movie

A movie created in Director that shows how the software for connecting to the Exploring Multimedia BBS works.
Exploring Multimedia BBS
Contains the software for calling the BBS.

Getting Started with TF
A text document and a Microsoft Word document that explain how to use the software.

TeleFinder/User
This folder contains the software. Drag it to your hard disk. Note that you don't have to have the software to call the BBS—it’s only necessary if you want to use the GUI interface. You can call using regular telecom software. The number of the bulletin board is (617) 666-9447.

TF Prefs 1 bit
This file contains a 1-bit version of the splash screen. Good for use on portables or other machines with only 1-bit screens. Simply drag the TF Prefs file from this folder into the TeleFinder/User folder.

Library
This folder contains images, QuickTime movies, and sounds that you can use in your own productions. While you can use these materials for your own purposes, they are not in the public domain and cannot be given or sold to others. They cannot be distributed as part of another collection or library of material.

Images
This folder contains 3 folders. Each folder contains the same set of twenty images; one folder contains 24-bit versions of the files, while the other folders contain versions that use the 8-bit system palette and 8-bit adaptive palettes, respectively.
QuickTime clips

This folder contains three folders of clips. The 180 x 135 and 240 x 180 folders contain the same movies in different sizes. All are compressed using the Apple Video Compressor and are 15 frames per second.

The folder 320 x 240 contains the same file, compressed once at normal quality and once at low quality using the Apple Video Compressor.

Sounds

Contains two folders that have the same sounds in Sound Edit and AIFF format (both at 22 kHz).

Book chapters

This folder contains elements relating to specific chapters in the book.

Graphics

Contains files related to Chapter 7, "Graphics."

Pictures

Sample images referenced in the chapter.

Photo CD stuff

Files from a Photo CD.

PCD1642 Slide Show

A QuickTime movie that shows the images contained on a PhotoCD.
Photos
This folder resembles the folder you see when viewing a Photo CD using QuickTime 1.5+. A Photo CD contains the same images in five different formats. This folder contains the five folders, and in each folder there are three sample images. You must be running QuickTime 1.5 (or greater) to be able to view these.

GIFConverter
Shareware program. Use for viewing and converting GIF images.

JPEGView
Shareware program. Use for viewing and converting JPEG images.

PICTuresque
Shareware program. A DA for viewing and printing graphics.

PICTshow
Public domain. Program for playing PICTs, listening to sounds, and playing QuickTime movies.

Animation
Contains an animation created using the program Life Forms. (See Chapter 8, “Animation.”)

3D
Contains a test image rendered with different qualities. (See Chapter 9, “3D.”)

QuickTime
Contains files related to Chapter 10, “QuickTime.”
Image Test
This folder contains the same image, compressed using the different QuickTime compressors.

Movie Test
This folder contains the same movie compressed using the different QuickTime compressors.

Shutdown Movie
Shareware program. Use to play a QuickTime movie when your computer is shut down.

Theater Maker
Shareware program. Use to add a background image to a QuickTime movie.

Sound
Contains files related to Chapter 11, “Sound.”

Sound Quality
A demo program which provides an example of a sound sampled at different rates.

Metro Demo
Demo version of the Metro MIDI sequencer.

Sound Converter
Shareware program. Use to convert sounds from one format to another.
SoundRoundUp
Shareware program. Use to extract sounds from applications and files.

Workshop
Contains the files necessary for creating the interactive newsletter project in Chapter 16, “Putting it to work.”
Also contains a working version of the project.
INDEX

MACINTOSH
Multimedia
WORKSHOP
Symbols

128M optical disk drives, 81
16 and 24 bit video graphics cards, 78-79
24STV (RasterOps) video digitizing boards, 193-195
3D applications, 166-167
Alias Sketch!, 173-174
CAD programs, 162
categories, 162-166
Electric Image, 174
general purpose programs,
165-166
hardware requirements,
174-175
Infini-D (Specular International), 170
MacroMind Three-D, 171-173
MacroModel, 174
modeling, 145-150
Ray Dream Designer, 167-169
ShowPlace, 174
StrataVision 3D (Strata Inc.),
171
68020/68030 processors, 72
68040 processors, 76

About... screens, HyperCard project, 318-320
Accelerator program (Director),
272, 276
Action! (Macromedia), 142, 245
Edit Object dialog box, 285
Scene Sorter window, 284-285
scenes, 284-286
Time window, 285

presentation program, 284-286
QuickTime movies, 197
ADDmotion (Motion Works),
136-138
Adobe
PostScript fonts, 90-91
Premiere, 191
Type Manager (ATM), 90-91
Type On Call, 91
After Effects, 204
Aldus Fetch, 231-235
Alias Research, Sketch! Sketch!,
149-150, 173-174
ambient lights, 157-158
AMIX electronic network, 238
animation, 144-145, 160-162
3D, 129
actors, 136-138
cartoon
anticipation, 129-131
squash and stretch, 130-131
cels, 136-138
Director, 277-278
dope sheets, 141-142
event-driven, 161
exporting formats
PICS/PICT files, 161-162
QuickTime movies, 161-162
key frame, 134-136, 160-161
memory management, 132
principles, 129-134
programs, 134-142
Action! (Macromedia), 142
ADDmotion (Motion Works), 136-138
Animation Stand (Linker Systems), 141-142
Animation Works (Gold Disk Software), 138-139
Cinemation, 139-140
CorelMOVE (Corel Corp.),
136-138
FilmMaker (Macromedia), 141-142
Magic (Macromedia), 142
MovieWorks, 140-141
PRMotion (Motion Works), 136-138
run-length encoding, 134
sound synchronization, 131-132
transferring PICS/PICT format, 133-134
video differences, 182-183
Animation compressors, 189-190
Animation Stand (Linker Systems), 141-142
Animation Works (Gold Disk Software), 138-139
Apple
AppleScript, 267
Macintosh
Human Interface Guidelines Companion, 45-46
OnLine Reference, 30-35
QuickTime Intro News, 35-39
Titles Sampler, 39-43
TrueType, 91
applications, standalone (SuperCard), 268-270
architectural modeling programs, 162-164
Astound! (Gold Disk), 282
asymmetrical compressors, 189
AT&T Graphics, MacTopas, 175
audio
CD integration, 223-224
compression, MACE (Macintosh Audio Compression and Expansion), 213
computer-synthesized speech (MacinTalk), 226
copyright laws, 235
digitizing/sampling, 213-217
editing, 218-219
files, snd (sound resources), 219-220
hardware, 210-213
improving sound quality, AudioMedia II card (DigiDesign), 223
programs
DECK (DigiDesign), 223
MIDI, 224-226
System 7.0, 214
recording
beep sounds, 216-217
MacRecorder (Macromedia), 214-217
problems, 212, 222
techniques, 221-222
sampling rates, 212-213
sequencers, MIDI (Musical Instrument Digital Interface), 224-226
utilities
Sound Converter Pro (David Lambert), 220
Sound RoundUp (Robert Gibson), 220
Audio Interchange File Format (AIFF), 220
Audio Visual Group, Multiple Media Tour, 43-45
AudioMedia II card (DigiDesign), 223
AudioShop (Opcode), editing audio, 218-219
authoring environments
Director (Macromedia), 245, 271-278
HyperCard (Apple), 245, 251-267
SuperCard (Silicon Beach Software), 245, 268-271
Authorware (Macromedia), 279-280
B

BackBurner (Specular International), 170
Background command, Edit menu, 300-303, 308-311
Background command, Go To menu, 257-258
background fields, 308-311
backgrounds (HyperCard), 257-260, 299-303
Bernoulli drives, 80-81
Bézier curves, 106
bitmap-based graphics programs, 102-108
bitmapped fonts, 88-91
Bkgnd Info... command, Objects menu, 259-260, 299-300
bounding boxes, 152
bump mapping, 155-156
Button Info... command, Objects menu, 294
button parameter dialog box, 261
button parameter dialog box, 298, 302-303, 317-318
button parameter dialog box, 261-263, 298-299
buttons, 99, 260-263

CD-ROM
clip collections, 231-234
contents of included disc, 322-332
drives, 81-83
single/multi-session, 123-124
Type On Call (Adobe), 91
CDs, movie integration, 223-224
cels, 136-138
Centris 610/650, 77-78
channels (Director), 274-275
Cinemation, 139-140
clip collections, 231-234
Clip Time (Alpha Technologies Group, Inc.), 233-234
ClipMedia, multimedia clip collections, 231-232
clipping, audio recording, 222
clock rates, 75-76
color, computer constraints for developing projects, 22
color graphics, 296-297
ColorizeHC, 307-312
ColorStudio (Letraset), 109-110
commands (HyperCard)
Background (Edit menu), 300-303, 308-311
Background (Go To menu), 257-258
Bkgnd Info... (Objects menu), 259-260, 299-300
Card Info... (Objects menu), 294
Go To, (Go To menu), 261-263
New Background (Edit menu), 259
New Background (Objects menu), 299-300

C

CameraMan (Vision Software), 201-202
Canvas (Deneba), 108-110
card fields, 256-257
Card Info... command, Objects menu, 294
Card Info... dialog box, 295
copyright laws, 235-237
fair use, 237
personal works, 235-237
sound works, 235
works for hire, 236
works of others, 236-237
CorelMOVE (Corel Corp.),
136-138
cost constraints, 18-19
customized programs, 246

data, planning project development, 28
DECK (DigiDesign), 223
decompressing QuickTime,
187-188
defaults, fonts, 301
delivery
constraints for developing
projects, 24
final, 52
post-delivery, 53
demonstrations, 283
desktop publishing, 9
developing
projects
constraints, 18-26, 61-62
final versions, 64-66
ideas/methods, 16-18
planning/reviewing, 26-50,
57-61
prototypes, 50-52
dictionaries, HyperCard project,
320
digital signal processors (DSPs),
227
digital sound processors, 84
digital video, 181-182
Digital Video Library (Educorp), 233-234
digitizing audio, 217
digitizing hardware acceleration programs
  Digital Film (SuperMac), 196-197
  MoviePak (RasterOps), 196-197
Director (Macromedia), 245, 271-278
  Accelerator program, 276
  advantages/disadvantages, 276-277
  animation, 277-278
  Cast window, 273
  components, 272-277
  factories, 278
  instruction, 278
  MIDI playback, 224-226
  Paint window, 273
  QuickTime movies, 197
  Score window, 274
  Text window, 273
  Windows Player program, 272
  XObjects, 278
disc, contents of included
  CD-ROM, 322-332
DiVA VideoShop, 191
documentation, 52
dope sheets, 141-142
DreamNet, Ray Dream Designer, 159-160
drivers, audio, MacRecorder (Macromedia), 217
drives, CD-ROMs, 81-83
DXF format, 150
DynaPerspective (Dynaware), 162-164

E
Edit Object dialog box (Action!), 285
editing
  audio, programs, 218-219
  documents, 268-270
  QuickTime movies, 198-204
Electric Image, 174
environments, multimedia, 242-247
EPS (Encapsulated PostScript file) format, 107-108, 113
Eraser tool, Tools palette, 253-257
error messages
  loading PICT, 310-311
  loading PICT resource dialog box, 296
event-driven animation, 161
extrusion, 146-147
extrusion modeling, 164-165

F
factories, 278
fair use, copyright laws, 237
Field dialog box, 255
Field editing tool, Tools palette, 301
Field Info... dialog box, 301, 311, 316-317
Field tool, Tools palette, 253-257, 308-309
fields, 253-257
files
  audio
    Audio Interchange File Format (AIFF), 220
    snd (sound resources), 219-220
DXF format, 150
EPS (Encapsulated PostScript file), 107-108
graphics formats, 111-117
JPEG (Joint Photographic Experts Group), 108
PICS (Picture), 104-106, 161-162, 307
QuickTime, 184-185, 198-200
RIB, 158-159
TIFF (Tagged Image File Format), 105-106
Workshop (CD-ROM disc), 292-294
film libraries, 230-231
FilmMaker (Macromedia), 141-142
flat shading, 152
Floating Point Units (FPUs), 73-76
font dialog box, 256, 304
Fontographer, 99
fonts, 96-99
  bitmapped, 88-91
copying, 98
creating, 98-99
default, 301
FontStudio, 99
laser, 90-91
point sizes, 88-91
PostScript (Adobe), 90-91, 97-99
  TrueType, 97-99
formatting text, 306-307
Fractal Designs Painter, 110, 124-125
frames per second (fps), 192-193
freeform modeling
  Infini-D (Specular International), 148-149
  isometric views, 148-149
ModelShop (Macromedia), 149-150
Sketch! Sketch! (Alias Research), 149-150
Swivel 3D, 148-149
FreeHand (Aldus), 106-111

G

GIF (Graphics Interchange Format), 114
Go To command (Go To menu), 261-263
Gold Disk Software, Animation Works, 138-139
Gouraud shading, 153
graphics
  color, 296-297
  file formats, 111-117
  MacDraw, 105
  PICT files, 307
  positions, 312
  pressure-sensitive drawing tablets, 124-125
programs
  bitmap-based, 102-108
  Canvas, 108-110
  ColorStudio, 109-110
  Fractal Designs Painter, 110
  FreeHand, 106-111
  Illustrator, 106, 109-111
  MacDraw/MacDraw Pro, 104-110
  MacPaint, 103-108
  Microsoft Excel, 111
  object-oriented, 102-108
  Photoshop, 109-110
  SuperPaint, 105-106, 110
Graphics compressors, 189-190
H

hard disks, 79-80
hardware
  accelerators, 160
  audio, 210-213
  constraints for developing projects, 19-24
digitizing
  Digital Film (SuperMac), 196-197
  video digitizing boards, 191-197
requirements for 3D applications, 174-175
sound digitizing, 213-217
Help screens, HyperCard project, 318-320
Hewlett-Packard Scanjet IIc scanner, 121
HyperCard, 245
  backgrounds, 257-260,
  299-303, 308-311
  buttons, 260-263
  cards, 250-252, 256-257
  ColorizeHC program, 307-312
  editing sounds, 218-219
  fields, 253-257
  file formats, snd (sound resources), 219-220
  hypertext, 92-96
  memory size, 293-294
  project, 318-320
  Recent window, 95-96
  script compilation,
    Compilelt! (Heizer), 263-267
Script window, 262-263, 294
stacks, 251-256
text, 252-253, 303-307
titles, 303-304
XCMDs, 266
XFCNs, 266
HyperCard Player (Claris), 268-269
HyperTalk, 92-96, 238, 263-265
message window (Command-M), 317
programming, 266-267
script conversion, Compilelt!
  (Heizer), 266
I

icons, 99
Illustrator (Adobe), 106, 109-111
ImageWriter printer, 103
Infini-D (Specular International), 144-145, 170
event-driven animation, 161
flat shading, 152
freeform modeling, 148-149
general purpose 3D applications, 165-166
Gourand shading, 153
lathing, 147
network rendering engines, 159-160
Phong shading, 153
Ray tracing, 154
wireframes, 152
interactive
  brochures, 283
  presentations, 245
isometric views, 148-149
J-K

JPEG (Joint Photographic Experts Group) format, 108,
114-117
key combinations
adding fields (Command key+click+drag), 308-309
Background command (Command-B), 300-301, 309-311
changing fonts (Command-T), 304
cursor changing to cross-hair (Command key), 298
HyperTalk message window (Command-M), 317
New Card, Edit menu (Command-N), 294, 310-312
New Card, Objects menu, (Command-3), 300
opening script (Option+Command+click), 302-303
previous card (Command-2), 300-303
key frame animation, 160-161
kiosk applications, 283
Kodak DCC-200 still video cameras, 119-120

languages, programming, 106-108
laser disks, 83-84
laser fonts, 90-91
LaserWriter printers, 106
lathing, Infini-D (Specular International), 147
Life Forms (Macromedia), 134-136
lighting, 157-158
Lingo, programming language, 272

link definitions (Special Delivery), 287-288
Linker Systems, Animation Stand, 141-142
linking (Action!), 286
loading PICT error message, 310-311
loading PICT resource dialog box error message, 296
lossy compression format, 108
Lotus, MarketPlace Business, 56-66

MacDraw, 104-108
MacDraw Pro (Claris), 108-110
MACE (Macintosh Audio Compression and Expansion), 213
MacinTalk, 226
Macintosh audio hardware, 210-213
Macintosh history, 70-76
Macintosh Human Interface Guidelines Companion, 45-46
Macintosh OnLine Reference, 30-35
MacPaint, 103-108, 111-112
MacRecorder, 186-187, 216-217
MacRecorder (Macromedia), 214-217
MacRenderMan (Pixar), 144-145, 154-155, 158-159
Macromedia
Action!, 142
FilmMaker, 141-142
Life Forms, 134-136
Magic, 142
ModelShop, 149-150, 162-164

Kodak DCC-200 still video cameras, 119-120

L

languages, programming, 106-108
laser disks, 83-84
laser fonts, 90-91
LaserWriter printers, 106
lathing, Infini-D (Specular International), 147
Life Forms (Macromedia), 134-136
lighting, 157-158
Lingo, programming language, 272
Macintosh Multimedia Workshop

MacroMind Three-D, 145, 171-173
  bump mapping, 155-156
  general purpose 3D applications, 165-166
  texture mapping, 155-156
MacroModel, 174
MacTopas (AT&T Graphics), 175
Macworld magazine, source materials, 231
Magic (Macromedia), 142
MarketPlace Business, 56-66
materials constraints, 19-20, 61-62
matting, 171-173
media
  cataloging utilities, 234
  copyright laws, 235-237
  film libraries, 230-231
  personal likenesses, 237
  QuickTime support, 179-184
  sources for materials
    Macworld magazine, 231
    New Media magazine, 231
  stock houses, 230-231
Media Cataloger (Interactive Media Corp.), 234
MediaGrabber, 193-194
MediaTime board (RasterOps), 195
memory
  management, animation, 132
  multimedia requirements, 77-84
  size for HyperCard, 293-294
message window (HyperTalk), Command-M, 317
Metro (OSC) MIDI programs, 224-226
Microsoft Excel, 111
MIDI (Musical Instrument Digital Interface), 224-226
modeling, 144-150
  architectural, 162-164
  extrusion, 146-147, 164-165
  StrataVision, 145
ModelShop (Macromedia), 149-150, 162-164
Motion Works, 136-138
Movie Player (Apple), 185, 197
Movie Recorder (Apple), 191
MoviePak (RasterOps), 196-197
MoviePlay, 201-202
movies
  QuickTime, 185
  recording, 186-191
MovieWorks, 140-141
Multi Media Music (Filler Tracks), 234
multi-pass video capturing, 194-195
multi-session Photo CDs, 123-124
multimedia, 8-9
  cataloging, Aldus Fetch (Aldus), 231-232
  Centris 610/650, 77-78
  clip collections, 231-234
  Clip Time, 233-234
  ClipMedia, 231-232
  Digital Video Library, 233-234
  environments, 242-247
  Macintosh LC III, 77-78
  Multi Media Music, 234
  Multiple Media Tour, 43-45, 232-238
  performance improvement, 78
  producing, 10-11
  Quadra, 77-78
  QuickTime, 246
  requirements, 77
  storage requirements, 79-84
  uses, 9-10
  WraptureReels One, 233-234
multiple windows, SuperCard, 268
network rendering engines, 159-160, 170
networks, AMIX, 238
New Background command (Edit menu), 259
New Background command (Objects menu), 299-300
New Card command (Edit menu), 252, 258, 294-295, 310-312
New Media magazine, source materials, 231
Newsletters, 292
Next command (Go menu), 294-295
noise, 190
None compressors, 190-191
object-oriented graphics programs, 102-108
objects, 104
Ofoto (Light Source, Inc.), 121
Opcode, MIDI program, 224-226
optical disk drives, 80-81

Paint bucket tool, Tools palette, 294-295
Paint window (Director), 273
Pascal, 263
Passport, MIDI program, 224-226
Passport Producer, 244
Passport Producer (Macromedia), MIDI playback, 224-226
Persuasion (Aldus), 197, 243-244
Phong shading, 153

Photo CD (Kodak), 122-125
Photo JPEG compressors, 189-190
photo-realism rendering algorithms, 151-160
Photoshop, scanners and, 121
Photoshop (Adobe), 109-110
PICS files, 133-134, 161-162
PICT files, 104-106, 112, 133-134, 161-162, 307
Picture Compressor (Apple), 115-117
Pixar
MacRenderMan/ShowPlace, 144-145
Typestry, 164-165
pixels, 23, 102
planes, 148-149
planning project development, 26-50
Player program (Director), 272
point lights, 157-158
point sizes, 88-91
Popcorn, 185
portals (Special Delivery), 287
PostScript (Adobe), 90-91, 106-108
Bézier curves, 106
fonts, 97-99
PowerPoint (Microsoft), 243-244
Premiere 2.0 (Adobe), 200-204
presentation programs, 282-283
Astound! (Gold Disk), 282
Special Delivery, 286-288
pressure-sensitive drawing tablets, 124-125
Previous command (Go To menu), 252, 262
printers
ImageWriter, 103
LaserWriter, 106
procedural shaders, 154
processors, 72, 75-76
programming languages
  HyperTalk, 263-267
  Lingo, 272
  Pascal, 263
  PostScript, 106-108
  QuickDraw, 106-108
programs
  Accelerator (Director), 272, 276
  Action! (Macromedia), 284-286
  Adobe Premiere, 191
animation
  Action! (Macromedia), 142
  ADDmotion (Motion Works), 136-138
  Animation Stand (Linker Systems), 141-142
  Animation Works (Gold Disk Software), 138-139
  Cinemation, 139-140
  CorelMOVE (Corel Corp.), 136-138
  Director, 128
  FilmMaker (Macromedia), 141-142
  Magic (Macromedia), 142
  MovieWorks, 140-141
  PROmotion (Motion Works), 136-138
  QuickTime, 132
audio
  DECK (DigiDesign), 223
  Sound Edit (Macromedia), 214-217
  Sound Quality, 212-213
authoring environments
  Authorware
    (Macromedia), 279-280
  Director, 245
  HyperCard, 245
  SuperCard, 245
capturing movies
  Screen Play, 192-193
  ColorizeHC, 307-312
Director Windows Player, 272
DiVA VideoShop, 191
documents
  SuperEdit (SuperCard), 268-270
  Fontographer, 99
  FontStudio, 99
graphics, 102-108
  Canvas (Deneba), 108-110
  ColorStudio (Letraset), 109-110
  Fractal Designs Painter, 110
  FreeHand (Aldus), 106-111
  Illustrator (Adobe), 106, 109-111
  MacDraw, 104-108
  MacDraw Pro (Claris), 108-110
  MacPaint, 103-108
  Microsoft Excel, 111
  Photoshop (Adobe), 109-110
  SuperPaint (Aldus), 105-106, 110
  HyperCard, 250-267
interactive presentations
  Action! (Macromedia), 245
  Special Delivery (TITM), 245
MacinTalk, synthesized
speech, 226
MediaGrabber, 193-194
MIDI
  Director (Macromedia), 224-226
  Metro (OSC), 224-226
  Opcode, 224-226
  Passport Producer (Macromedia), 224-226
Movie Play, 185, 201-202
Movie Recorder, (Apple), 191
movies
  Premiere, 200
  VideoShop, 200
Player (Director), 272
Popcorn, 185
presentation, 282-283
Prograph, 279-280
QuickDraw GX, 71-72, 185-186
ResEdit, 98-99
scanning images
  Hewlett-Packard Scanjet IIc, 121
  Ofoto (Light Source, Inc.), 121
slide presentations
  Passport Producer, 244
  Persuasion (Aldus), 243-244
  PowerPoint (Microsoft), 243-244
SuperCard, 268-270
video digitizing acceleration
  Digital Film (SuperMac), 196-197
  MoviePak (RasterOps), 196-197
see also authoring environments
project windows, 268-270

projects
  Macintosh Human Interface Guidelines Companion, 45-46
  Macintosh OnLine Reference, 30-35
  Multiple Media Tour, 43-45
  QuickTime Intro News, 35-39
  SuperCard, 268-270
  Titles Sampler, 39-43
PROmotion (Motion Works), 136-138
prototypes, 50-52, 57-61, 64-66
publications, 292, 303-307

QuickDraw GX, 71-72, 91, 106-108, 185-186
QuickTime
  animation programs, 132
  compressing, 183-184, 187-190
  computer constraints for developing projects, 21
decompressing, 187-188
digital video storage, 179-181
editing movies, 198-200
  Premiere 2.0 (Adobe), 202-204
  Sparrow (DiVA/Avid), 204
  VideoShop 2.0 (DiVA), 202-204
installing, 184-185
movie clips, 231-234
movies, 161-162, 185, 197-202
multimedia uses, 246
recording movies, 186-191
referencing movie files, 198-200
special effects, 204
supporting media, 179-184
 timing mechanism, 183-184
 utilities, 200-205
 version 1.6 CD integration, 223-224
 Video For Windows (Microsoft), 205-206
 QuickTime for Windows (Apple), 205-206
 QuickTime Intro News, 35-39

---

R

Radius Rockets, 160
RasterOps
 MediaTime board, 195
 video digitizing boards, 118-119, 193-195
 Ray Dream Designer, 144-145, 167-169
 3D applications, 165-166
 extrusion, 146-147
 network rendering engines (DreamNet), 159-160
 texture mapping, 155-156
 Ray tracing, 154
 Raydiosity, StrataVision (Strata Inc.), 171
 Recent window, 95-96
 recording
 audio
 problems, 222
 quality, 212
 techniques, 221-222
 reflection maps, 156
 releases, 237
 rendering, 144-145, 151-160
 RenderMan RIB format (Pixar), 158-159

---

S

sampling
 audio, 217, 221
 rates, 212-213
 scanned images, 120-121
 Scene Sorter window (Action!), 284-286
 Score, Director components, 272-277
 Screen Play, 192-193
 screens, 23, 294-295
 About... (HyperCard project), 318-320
 Help (HyperCard project), 318-320
 script execution, 263-265
 Script window, 252-263, 294
 selecting computers, 76
 Serious, 279-280
 shading, 151-154
 ShowPlace (Pixar), 144-145, 174
 Shutdown Movie, 204-205
 single-session Photo CDs, 123-124
 Sketch! Sketch! (Alias Research), 149-150
 slide presentations, 243-244
 Small Computer System Interface (SCSI), 71-72
 snd (sound resources) files, 219-220
 Sony Mavica still video camera, 119-120
 sound compression, 213
Sound Converter Pro (David Lambert), 220
sound digitizing, 213-217
Sound Edit (Macromedia), 214-217
audio editing, 218-219
audio files, 219-220
sampling, 221
sound level problems, 222
Sound Quality, 212-213
sound quality, 222-223
Sound RoundUp (Robert Gibson), 220
sound synchronization, 131-132
sources for project ideas
 Macintosh Human Interface Guidelines Companion, 45-46
 Macintosh OnLine Reference, 30-35
 Multiple Media Tour, 43-45
 QuickTime Intro News, 35-39
 Titles Sampler, 39-43
Sparrow (DiVA/Avid), 204
Special Delivery (ITC)
 link definitions, 287-288
 portals, 287
 presentation program, 286-288
 QuickTime movies, 197
special effects, 204
Spectator (Baseline Publishing), 201-202
Specular International, Infini-D, 144-145, 148-149
speed of playback, 22
spotlights, 157-158
squash and stretch cartoon animation, 130-131
stacks, 251-252
Stage, Director components, 272-277
standalone applications, SuperCard, 268-270
step-and-grab video capturing, 194-195
still video cameras
 Kodak DCC-200 series, 119-120
 Sony Mavica, 119-120
stock houses, 230-231
storyboards, 47-48
StrataVision (Strata), 144-145
3D applications, 165-166
lighting, 158
network rendering engines, 159-160
StrataVision 3D (Strata Inc.), 148, 171
streaming, 183-184
SuperCard (Silicon Beach Software), 245, 268-271
 hypertext, 92-96
 multiple windows, 268
 projects, 268-270
 uses, 270
SuperEdit (SuperCard), 268-270
SuperEdit project window, 268-270
SuperPaint (Aldus), 105-106, 110
surfaces, 154-156
sustainable transfer rate, 79-80
sweeping, 148
Swivel 3D, 148-149
symmetrical compressors, 188-189
SyQuest drives, 80-81
System 7.0 audio applications, 214
<table>
<thead>
<tr>
<th>T</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>tape recorders, 221</td>
<td>user interface, 28-29</td>
</tr>
<tr>
<td>Targa, 113</td>
<td>utilities, 200-205</td>
</tr>
<tr>
<td>television recording, 187</td>
<td>audio</td>
</tr>
<tr>
<td>testing prototypes, 62-64</td>
<td>Sound Converter Pro</td>
</tr>
<tr>
<td>text, 252-253</td>
<td>(David Lambert), 220</td>
</tr>
<tr>
<td>copying, 305-306</td>
<td>Sound RoundUp (Robert</td>
</tr>
<tr>
<td>formatting, 306-307</td>
<td>Gibson), 220</td>
</tr>
<tr>
<td>locking fields, 316-317</td>
<td>media cataloging</td>
</tr>
<tr>
<td>wrapping manually, 313-314</td>
<td>Aldus Fetch (Aldus),</td>
</tr>
<tr>
<td>Text tool, Tools palette, 252-257,</td>
<td>234-235</td>
</tr>
<tr>
<td>303-304</td>
<td>Media Cataloger (Interactive Media Corp.), 234</td>
</tr>
<tr>
<td>Text window (Director), 273</td>
<td>Picture Compressor (Apple),</td>
</tr>
<tr>
<td>texture mapping, 155-156</td>
<td>115-117</td>
</tr>
<tr>
<td>TheatreMaker, 204-205</td>
<td>screen capturing, 201-202</td>
</tr>
<tr>
<td>Three-D (MacroMind), 145</td>
<td>Shutdown Movie, 204-205</td>
</tr>
<tr>
<td>TIFF (Tagged Image File Format), 105-106, 112</td>
<td>TheatreMaker, 204-205</td>
</tr>
<tr>
<td>time constraints, 19</td>
<td></td>
</tr>
<tr>
<td>Time window (Action!), 285</td>
<td></td>
</tr>
<tr>
<td>timing mechanisms, 183-184</td>
<td></td>
</tr>
<tr>
<td>titles, HyperCard, 303-304</td>
<td></td>
</tr>
<tr>
<td>Titles Sampler, 39-43</td>
<td></td>
</tr>
<tr>
<td>Tools palette (HyperCard)</td>
<td></td>
</tr>
<tr>
<td>Browse tool, 254-257, 262,</td>
<td></td>
</tr>
<tr>
<td>299, 305-306, 309-311</td>
<td></td>
</tr>
<tr>
<td>Button tool, 261-263, 298-299</td>
<td></td>
</tr>
<tr>
<td>Eraser tool, 253</td>
<td></td>
</tr>
<tr>
<td>Field editing tool, 301</td>
<td></td>
</tr>
<tr>
<td>Field tool, 253-257, 308-309</td>
<td></td>
</tr>
<tr>
<td>Paint bucket tool, 294-295</td>
<td></td>
</tr>
<tr>
<td>Text tool, 252-257, 303-304</td>
<td></td>
</tr>
<tr>
<td>training systems, 283</td>
<td></td>
</tr>
<tr>
<td>transparency maps, 156</td>
<td></td>
</tr>
<tr>
<td>TrueType fonts (Apple), 91, 97-99</td>
<td></td>
</tr>
<tr>
<td>type families, 96-99</td>
<td></td>
</tr>
<tr>
<td>Type On Call (Adobe), 91</td>
<td></td>
</tr>
<tr>
<td>typefaces, 96-99</td>
<td></td>
</tr>
<tr>
<td>Typestry (Pixar), 146-147,</td>
<td></td>
</tr>
<tr>
<td>164-165</td>
<td></td>
</tr>
<tr>
<td>video</td>
<td></td>
</tr>
<tr>
<td>animation differences, 182-183</td>
<td></td>
</tr>
<tr>
<td>compressing, 190-191</td>
<td></td>
</tr>
<tr>
<td>digital, 181-182</td>
<td></td>
</tr>
<tr>
<td>digital storage, 179-181</td>
<td></td>
</tr>
<tr>
<td>Photo CD (Kodak), 122-125</td>
<td></td>
</tr>
<tr>
<td>recording images, 118-119</td>
<td></td>
</tr>
<tr>
<td>scanned images, 120-121</td>
<td></td>
</tr>
<tr>
<td>still images, 119-120</td>
<td></td>
</tr>
<tr>
<td>television recording, 187</td>
<td></td>
</tr>
<tr>
<td>video compressors, 188-191</td>
<td></td>
</tr>
<tr>
<td>video digitizers</td>
<td></td>
</tr>
<tr>
<td>RasterOps 24MXTV, 118-119</td>
<td></td>
</tr>
<tr>
<td>RasterOps 24STV, 118-119, 193-195</td>
<td></td>
</tr>
<tr>
<td>VideoSpigot (SuperMac), 118-119, 186-207</td>
<td></td>
</tr>
</tbody>
</table>
VideoVision (Radius), 118-119, 186-187, 195-196
video display boards, 180
Video For Windows (Microsoft), 205-206
video graphics cards, 78-79
VideoFusion, 204
VideoShop 2.0 (DiVA), 200-204
Virtus, Walkthrough, 163-164

W

Walkthrough (Virtus), 163-164
windows, 262-263, 268
wireframes, 152
works for hire, copyright laws, 236
works of others, copyright laws, 236-237
Workshop files (CD-ROM disc), 292-294
WraptureReels One (Form & Function), 233-234

X-Y-Z

Xanadu (Ted Nelson), 238
XCMDs, 266, 293-294
XFCNs, 266
XObjects (Director), 278
YARC Company, 160
About the CD-ROM

The CD-ROM included with this book is in HFS (Macintosh) format, and includes a wide range of things—from demos and sample files to images and QuickTime movies. (Note that some of the software is shareware, and is provided to you on a trial basis.) For more information about the disc contents, see the “What’s on the disc” appendix.

Different applications and files on the disk have different requirements; we recommend a color Macintosh with System 7 and 4M of memory (and a CD-ROM drive!) as a minimum. Some of the applications require more memory, and some must be copied to your hard disk to work. The QuickTime movies must be viewed with an application that can play QuickTime movies (CameraMan’s MoviePlay and Adobe Premiere can do so.) You must also install the QuickTime extension, which is included on the disk. Some demos (and the “Putting it to work” files) require HyperCard.

The disc includes:

Tour the disc (a QuickTime movie)
QuickTime 1.6
Cover art
Software demos and samples
Includes demos of the following programs: Vividus’ Cinemation; Specular International’s Infini-D 2.0 (with sample images, QuickTime movies, and an interactive presentation); sample images and movies generated using Playmat; No Hands Software’s Common Ground (and demo documents); Virtus WalkThrough (with sample images and movies, and three Voyager documents that enable you to explore a 3D world); Heizer Software’s MasterScript, WindowScript, and CompileIt; Fractal Design’s Painter 2.0; a catalog for the Multiple Media Tour from Audio Visual Group; sample images generated using Strata 3D; Adobe’s Premiere 1.0, Photoshop 2.0, and Illustrator 3.0, and sample movies created using Premiere; Movie Works and sample presentations created with it; Gryphon Software’s Morph; Motion Works’ PRomotion; sample movies and an interactive presentation about VideoFusion; a demo movie created using VideoToolKit; and CameraMan’s MoviePlay utility.

Multimedia productions created by the author
Includes: the QuickTime Handbook movie; Now & Zen; Farmers Market movies; and Dogcow.

Exploring Multimedia
A monthly newsletter for Macintosh multimedia producers. On the disc there is an electronic copy of the first issue of the newsletter. There is also software for calling the Exploring Multimedia BBS.

Library
A folder that contains images, QuickTime movies, and sounds.

Chapters
Folder containing elements relating to specific chapters in the book: graphics (including sample images, Photo CD information, GIFConverter, JPEGView, Picturesque, and PICTshow); a sample animation; a sample 3D image; QuickTime materials (including samples, Shutdown Movie, and Theater Maker); sound materials (including a Sound Quality demo; a demo of the Metro MIDI sequencer, Sound Converter, and Sound Roundup); and files for the “Putting it to work” do-it-yourself chapter.
Make sizzling hot multimedia pop from your Mac!

CD-Rom includes:
More than 400M of dazzling video, searing graphics, audio, software demos, and more!

MACINTOSH Multimedia WORKSHOP
Create Multimedia Brilliance

Filled with powerful and professional multimedia design strategies for:

- Text, graphics, and animation
- Choosing and using various multimedia hardware tools
- QuickTime movies
- Sound and PhotoCD enhancements
- Working with multimedia environments, including authoring and presentations
- and more!

About the author

SKU - 926314
MACINTOSH MULTIMEDIA WORKSHOP W/DK

Used, Very Good