

Rochester Institute of Technology

RIT Digital Institutional Repository

Theses

5-23-1992

PresentationMaker graphic design archive module 3.2

Jane Settergren

Follow this and additional works at: <https://repository.rit.edu/theses>

Recommended Citation

Settergren, Jane, "PresentationMaker graphic design archive module 3.2" (1992). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the RIT Libraries. For more information, please contact repository@rit.edu.

Rochester Institute of Technology

A Thesis Submitted to the Faculty of
The College of Fine and Applied Arts
in Candidacy for the Degree of
MASTER OF FINE ARTS

PresentationMaker Graphic Design Archive Module 3.2

By Jane Ann Settergren
May 23, 1992

Approvals

Advisor: Professor James Ver Hague

Date:

5.19.92

Associate Advisor: Professor R. Roger Remington

Date:

May 19, 1992

Associate Advisor: Professor Mark Collien

Date:

19 May, 1992

Special Asst. to the Dean for Graduate Affairs: Professor Philip Bornarth

Date:

5/29/92

Acting Dean, College of Fine and Applied Arts: Dr. Peter Giopulos

Date:

6/2/92

I, Jane Ann Settergren hereby grant
permission to the Wallace Memorial Library of RIT,
to reproduce my thesis in whole or in part.
Any reproduction will not be for commercial use or profit.
Date:

Acknowledgments

I wish to thank my husband, Paul R. Settergren, and my Professors, Jim Ver Hague, Roger Remington, and Mark Collien, for their assistance and support on this project. I also wish to thank Professor Barbara Hodik for her support and encouragement as my client in this project.

Contents

1. Project Proposal	1
2. Review of Literature	
Interactivity	9
Hypermedia	10
Multimedia	11
Methodologies	15
Multimedia Software	15
Multimedia Hardware	19
Presentation Graphics	21
Evaluating Systems	23
Audio	24
Understanding Sound	26
Synthesized Sound	28
MIDI Interface	29
Video	30
Animation	34
Working with Images	36
3. Journal	39
4. Evaluation and Summary	81
Appendices:	
A: Time Schedules	
B: Flow Charts	
C: Palette Development	
D: PresentationMaker Screens	
E: Preparation of PICT Files	
F: Thesis Show Materials	
G: Scripts	
H: Glossary	
I: Bibliography	

Chapter 1: Project Proposal

1. Title: PresentationMaker
Graphic Design Archive Module 3.2

2. Client and Address:

Thesis Committee: Professor Jim Ver Hague
Professor Roger Remington
Professor Mark Collien

Faculty Client: Professor Barbara Hodik

Address: Rochester Institute of Technology
College of Fine and Applied Arts
1 Lomb Drive
Rochester, NY 14623

3. Designer: Jane Ann Settergren
5435 Thomas Rd.
Canandaigua, NY 14425

4. Thesis Statement:

My MFA Thesis project is to design and program an interactive presentation maker to be used by instructors in preparing material to be presented in a classroom setting.

5. Context of the Thesis:

This project stems from two separate sources. In the Spring of 1991, I was introduced to computer programs (Macintosh) that were capable of creating presentations from existing material stored on CD-ROM and laserdisc. The original concept for this thesis was to design an interactive media program that used this stored material and could be used by classroom instructors.

Also, at that same time period, I was asked to work on the Graphic Design Archive Project at Rochester Institute of Technology as my Master of Fine Arts thesis project. Given my experience in interactive media design (Certificate from American Video Institute), my desire to further explore and expand the dimensions of such programming along with Graphic Design Archive Project's expressed desire for further interactive development and capability, I agreed.

My interest and desire in interactive multimedia specifically designed to combine images and text resulting in a usable presentation for classroom teachers became the basis for my thesis project. In addition to that I would try to add the more specific dimensions desired in this particular project relating to the Graphic Design Archive.

6. Projected Assumptions about the User:

The intended user is the classroom instructor who has the desire to create an on-screen presentation. Many will have had experience with computers and be computer literate. However they may not be experienced with the Macintosh platform or with HyperCard software. There are tutorials that teach those skills and concepts; it is not within the realm of this thesis to provide that training.

PresentationMaker is a HyperCard program, and needs version 2.1 in order to run. HyperCard 2.0 cannot properly support all the scripts for the various windows. If the user wishes to add color pictures to the presentation, other software (eg. PhotoShop) and hardware (color scanner) may be needed to prepare the images. Subject matter expertise and organization for each presentation will be provided by the user.

Projected machine environment:

- Macintosh II computer with minimum of 5 MB RAM and 3 1/2" 1.4MB Internal Drives and hard drive
 - Apple color high resolution RGB 13" monitor with resolution of 640X480 pixels
 - Laserdisc player and monitor (optional)
 - LaserWriter Printer
- (Station projected to be available in the IMS Center)

7. Situation Analysis:

Images draw attention, illustrate ideas, and can provide explanation. Combining text with images can speak even more eloquently than the image alone. Computers have the capability to incorporate images and text easily. This provides the starting point for a presentation that would not only combine images and text, but could also include audio, video, and animation clips.

This combining of various media has popularly been termed "multimedia." A series of images could be created, combined with text and an appropriate background to create the audio-visual segments of a presentation. When combined with interactivity, the power and possibilities created for future use are intriguing.

Presently there exists a vast collection of images from the Graphic Design Archive stored in analog format on a videodisc. Along with these images, several modules have been developed to access these images in a variety of ways. Taking advantage of the accessing capabilities and combining these images with text in a usable format presents a challenge. If it is possible to do this, the images could be more available to instructors in their teaching or for their use in presentations.

8. Goal:

The intention of this thesis is to develop a tool for instructors that would assist them in the creation of a presentation in which a series of images could be used, such as from the Graphic Design Archive (GDA) videodisc.

9. Objectives and Considerations:

- a. to allow the user to create a visually interesting and informative presentation using the GDA archival images on the videodisc.
- b. to construct an application that is simultaneously functional (program does what it is intended to do), highly interactive (user makes the choices that determine the outcomes) and easy to use (eg. menu choices).
- c. to permit flexibility for the user in the arrangement of the images and the entering of text files for titles and handout information.
- d. to work harmoniously with the archive project, using the best and most appropriate tools for the existing project.
- e. to maximize the imagery on the videodisc, presenting the images in a visually interesting and aesthetically stimulating manner.
- f. to design an effective interface tool for the user that invites use as the purposes of the parts are clear and obvious to the user.
- g. to utilize a "multi-media" approach.

h. to apply the ten general principles of Human Interface Design as found in Apple's HyperCard Stack Design Guidelines.

1. The use of a clear metaphor with appropriate visual and audio effects can allow users to work with a set of existing expectations and apply those to the new situation.

2. Users need to be able to directly manipulate their environment, and they need visual and audio feedback.

3. It is much easier to "see-and-point" as opposed to "remember-and-type".

4. There needs to be consistency within the following:

- the graphic look
- the arrangement and grouping of buttons
- visual and audio feedback
- card layout
- background for cards with similar functions
- the stack structure

5. WYSIWYG or (What you see is what you get)

The layout of the stack should not be a puzzle to the user.

6. The user must have control of the actions, not the computer. The computer does the work, but the user must be the director of it.

7. Appropriate feedback and dialog allow the user to know what is going on, if a choice is inappropriate at a given time, etc. Feedback can allow the user to learn from his/her mistakes.

8. Because users generally make mistakes, forgiveness for them must be built into the program.

9. A perceived stability allows the user to feel comfortable and more in control than a program that seems to change at random.

10. Aesthetic integrity will enhance the effectiveness of the program.

i. to evaluate the module as it is developed and modify, incorporating changes, as appropriate.

10. Processes and Strategies:

- a. Collect information
 - 1. Research topic in library, using texts and periodicals
 - 2. Interview users.
 - 2.1 understand problems
 - 2.2 discover needs
 - 3. Understand existing project.
- b. Analysis of user needs
- c. Define project requirements
- d. Task analysis
 - 1. Break complex tasks into parts
 - 2. Define structure
 - 3. Set sequence of tasks
 - 4. Flowchart and diagram problem
- e. Create prototype
- f. Software development
 - 1. Programming
 - 2. Solicit user's input on continual basis
 - 3. Develop evaluation devices
- g. Final stages:
 - 1. Stringent testing with user and others
 - 2. Eliminate bugs
 - 3. Refine designs
 - 4. Finalize documentation

11. Time/Implementation:

<u>Week</u>	<u>Date</u>	<u>Tasks</u>
16	Dec 2	Clarify project ideas Time Line
15	Dec 9	Begin FlowChart Continue Research
14	Dec 16	Finish Research Work on Overall Layout
13	Jan 6	First Draft of Chapters 1 & 2 Work on importing Text
12	Jan 13	Chap.1 & 2 Revisions Gather data & images (Agha & Golden)
11	Jan 20	Write up of Journal to date Work on importing images (b/w bitmaps)
10	Jan 27	Revisions to overall layout, flowchart, etc. Work on importing color images
9	Feb 3	Continue Revisions Work on output methods
8	Feb 10	Write up of Journal to date Work on introduction and help
7	Feb 17	First major testing Revisions
6	Mar 9	Continue Revisions Write up of Journal to date
5	Mar 16	2nd major testing Begin other written parts of thesis
4	Mar 23	Revisions to program Continue writing
3	Mar 30	Final Evaluations and Recommendations Draft of Final Documentation
2	Apr 6	Last revisions and create final documentation
1	Apr 13	Prepare materials for show Documentation to printer

12. Available Resources:

Macintosh IIx computer with 8mb RAM
Videodisc and monitor for GDA disk
LaserWriter for hard copy output
Project targeted for completion by April 13th.

13. Dissemination:

A student station is presently available at the IMS Center for student use with the GDA project. It is projected that perhaps in the future another such station may be available for faculty.

14. Evaluation Plan:

Barbara Hodik has agreed to be the faculty client, giving input into the project at regular intervals. At the beginning of the project, the goals and objectives are to be set up. These goals and objectives will provide the basis against which the project can be evaluated on at regular periods.

Missing Page

Chapter 2: Review of Literature

Interactivity

Interactivity allows the users to shape their own experiences, generally manipulating 3 variables:

Frequency - How often the user can have input or is allowed to make choices?

Range - How many constraints are made on choices? Are the choices finite or infinite?

Personness - Does the user participate directly or indirectly? Is the user the actor or put in an observer role?

Reference: "Interactive Media and HyperCard" from Minds in Motion, Spring 1989.

Technological advances are sometimes viewed as decays in living standards rather than advances, with critics pointing out that there is increasing centralization and control of society and encouragement of passivity. A brighter view points toward increased opportunities for individual choice. Actually the technology itself guarantees neither, rather both capabilities are inherent in the tool, and its use will be determined by the user.

Creation of interactive works is somewhat unprecedented in art history and will need its audience to learn new conceptual and technical skills. The artist must create for the wide range of choices of the audience, and must have knowledge of the psychology of the audience and nature of the context. (p. 173)

In more traditional programming environments programs are self-running giving the user choices at specific times. Once started, the program moves through a series of preprogrammed steps. In this way, such a program is similar to a movie, a book, or a musical piece that has a certain order, and is, in a sense, linear.

In an interactive sequence, the user determines the sequence, and the order will be unique. The program needs to make sense and be engaging in a variety of orders. "The distinction between preprogrammed and user choice will undoubtedly become a major aesthetic design issue in the future." (p. 191)

Reference: MultiMedia Design with HyperCard by Stephen Wilson, Englewood Cliffs, NJ: Prentice Hall, 1991.

Large displays let you run a multimedia production in front of a large audience, but they create another problem: how do you control the presentation while standing far away from your Mac? Enter Proxima, with a system called Cyclops. You can aim its pointer at a large projection screen and use it to point and click just as if you were handling a mouse. A small video camera mounted on the LCD projection panel picks up the image of the pointer's LED positions on the screen. (\$1200)

Reference: "Multimedia and Video" by Lon McQuillin, MacUser, Feb. 1991, pp. 4-32 in Buyer's Guide Supplement.

While television has an uncanny ability to grab a viewer's attention, it remains a passive medium. The personal computer on the other hand, is a highly interactive tool, which until now has been pretty much restricted to searching through vast quantities of data and text. Thanks to the laser technology, videodiscs and compact music discs have become the cheapest method ever devised for storing information.

Today's machines offer aspects of the interactive multimedia experience, but none of them deliver anything close to the vision of the future: a device that could play music, movies, games, while providing viewers with a joy stick allowing them to pursue their own interests or needs.

Reference: "The World on a Screen" by Philip Elmer-Dewitt, Time, October 21, 1991, pp. 80-82.

Hypermedia

Interactive multimedia projects (sometimes called hypermedia) "can be viewed as electronic information webs on communities of knowledge." (p. 8) They use to advantage possibilities of videodiscs, CD ROM and Hypermedia software (ex. HyperCard). An interactive multimedia project allows for text, static pictures, graphics, video sequences, animation, simulation, music, and voice. In these systems, the design and navigation system becomes as important as the content. Also the speed that users can absorb and understand information "depends heavily on the presentation of the information." (p.9)

Reference: "The Glenn Gould Profile" by Jerry Durlak, Minds in Motion, Fall 1989, pp. 7-13.

Computers are bringing more revolutionary changes than merely working with several media simultaneously. In addition programmability (allowing user to make choices) can be built in, so that the user can customize the use to his/her own needs. This type of thinking is called Hypermedia; hyper meaning nonlinear method of moving through a body of information, and media referring to the variety of formats in addition to text (graphics, audio, video, and animated graphics). (p. 3)

Reference: MultiMedia Design with HyperCard by Stephen Wilson, Englewood Cliffs, NJ: Prentice Hall, 1991.

"Hypermedia can be defined as multimedia that has an interactive component to it." (p. 553) Examples are electronic encyclopedias, interactive maps, and games. Expert systems built around if-then-else equations use hypertextual construction. With optical storage devices and innovative video and sound compression techniques, the 90's will usher in a new generation of interactive hypermedia applications.

"Most hypermedia applications ship with a global system map or browser to show overall location, and then mark a trail as you navigate the system to aid you in finding previously viewed information." (p. 554) The software for multimedia is "like a conductor directing all the elements of a multimedia orchestration." The first popular hypermedia application on a personal computer was Bill Atkinson's HyperCard for the Macintosh released in 1988.

The introduction of text materials into electronic forms causes problems in two areas: design of the surface and design of the interfaces.

Surface design would include the typography, layout, graphics, illustrations, and the quality of the language.

Interface design refers to the problem of navigating through the text and is manifested on 3 levels: the immediate structure of the text (information provided within a page), the internal structure (information provided within the entire document), and the external structure (navigational aids that allow users to move from one document to another.)

On the page level, icons or symbols, menus, and color coding have all been used to aid the user. On the document level, where printed materials clearly have contents, indexes, appendices, and footnotes, users of electronic materials want help and look to the program to provide it.

It seems to be a critical element to understand how the user conceptualizes the system and material contained therein. Initial understanding is usually based on metaphors of similar aspects in the real world. DOS, for example, uses this approach with their "tree" file analogy. The interface must be carefully considered.

Reference: "Instructional Text: The Transition from Page to Screen." Stephen T. Kerr. Visible Language. Autumn 1986. pp. 368-392.

While HyperCard has spawned a tremendous quantity of custom programming by average users, there are few commercially successful products. Its primary value is as a multimedia developers' tool, and its success has been expanded on by SuperCard (Mac) and Plus (Mac, PC), and translated into Toolbook (Windows3).

Multimedia

Elements used in multimedia (text, graphics, sound, voice, music, video, and so on) are all extremely memory intensive. (p. 555) Manipulation of the raw files would be unwieldy, so multimedia authoring programs reduce elements to objects. (p. 556) Object-oriented programming (OOPs) defines a hierarchy of object classes so that user-generated actions or events (messages) can be handled in a logical manner. OOPs are not new, having arisen out of Xerox PARC in the 70's. OOPs are high level languages, and require much less code to create the same effect than a structured language, as C, Pascal, BASIC, or others. But they are slower to execute than the structured languages. (p. 557)

Reference: Beyond the Desktop: Tools and Technology for Computer Publishing. by Barrie Sosinsky, New York: Bantam Books, pp. 549 - 597.

The technology known as interactive multimedia encompasses a variety of systems for bringing information, music, voice, animation, photos, and video images together on a screen. This represents the convergence of three key communications technologies: television, personal computers, and laser storage systems, like the videodisc and the compact disc. Just like the beginning of other technologies, the road at first is bumpy, and littered with false starts, set backs, and skepticism.

Reference: "The World on a Screen" by Philip Elmer-Dewitt, Time, October 21, 1991, pp. 80-82.

Multimedia cont.

Is multimedia a battle between the printed word and “flashy visuals” and “sound bites”? It seems to have potential for both good and bad; in addition it can be technically complex and also quite expensive.

As an educational tool it has tremendous potential. Students are often encouraged to make their own documentaries based on videodisc images. This raises an important question—what are students being expected to learn here - are they learning about the subject matter, or how to create a documentary. The tools of creating, editing, and producing this become more of a focal point than the actual subject matter that the student is supposed to be focusing upon.

What is multimedia? It generally means using an authoring program to create and playback a production. It also generally takes a quantity of hardware (a Mac computer, a video camera or VCR, a CD Rom player, to mention a few) as well as numerous software packages to create all the pieces.

The most exciting and significant part of multimedia is the interactivity that is possible. A videodisc player works well in this area as it has random access, containing 54000 frames and almost immediate access to any of them. Frames can be played moving or as stills, and also audio can be part of the track.

Reference: “Getting Started with Multimedia” by Jim Heid, MacWorld, May 1991, pp. 225 -232.

Multimedia is a document that addresses more than one sense at a time, ex. a greeting card that plays a tune, or getting cash from an ATM machine, or use of a kiosk at a store. Interactive terminals are so accepted and have received so much attention that they have attracted most of the heavy weights of the computer industry: IBM, Apple, Commodore, Intel, Sony, Phillips, and many others. (p. 549)

Computer projects in multimedia have focused on:

- Training and education
- Sound and video production
- Presentations, designs, and simulations
- Art and entertainment
- Advertising

“Computers are uniquely suited to mix sight and sound to create living documents.” (p. 550) This technology is excellent for simulations and has been used to train airline pilots. Becoming more sensory inclusive and realistic we approach virtual reality experiences. “Virtual reality results when several of the senses are mixed to create an extremely realistic simulation.” (p. 551)

Multimedia presently receives tremendous press because it is a new frontier in communications; it is perceived as either a passing fad or a “watershed technology that will be as revolutionary as desktop publishing was.” (p. 551) Predictions are made of PCs controlling televisions and sound, replacing all the separate entertainment devices in homes today. However, multimedia is a “moving target” (p. 551) as it is dependent on

the fast changing capabilities of the computer and new techniques and technologies are being introduced at a fast pace.

There are two concepts that make computer multimedia projects unique as information mediums: "interactivity and nonlinear access." (p. 551) A computer program can be written so that the user controls the selection of an action or event. Studies have already established the value of involving several of the senses in communication. Recall is enhanced when more senses are involved, as in the saying people remember 20% of what they hear, 40% of what they see, and 60% of what they eat (interact with). (p. 552) A compound approach allows for more avenues for the presentation of material which makes them easier to learn and understand, and therefore more memorable. "Using a multipath approach is the best and most common way to learn and retain information." (p. 552)

Reference: Beyond the Desktop: Tools and Technology for Computer Publishing, by Barrie Sosinsky, New York: Bantam Books, pp. 549 - 597.

Multimedia includes media that allow us to input and combine art, animations, graphics, video-scanned images, sound and text into multi-sensory displays.

Reference: "Interactive Media and HyperCard" from Minds in Motion, Spring 1989

Though the cost of these programs is high, multimedia programs are likely to be enthusiastically received in America's schools. Some argue that we need this type of technology to teach a generation of children raised on Sesame Street and MTV; that we need to make education appropriate to the times we live in.

Others are not so sure. While multimedia can definitely prove useful in visualizing abstract concepts in science and math, the fear is that multimedia is basically to entertain at the cost of thinking. Multimedia can turn the traditional education experience into something more like TV, a world where words are unnecessary.

At the present interactive multimedia is a powerful tool whose best uses remain on the horizon.

Reference: "The World on a Screen" by Philip Elmer-Dewitt, Time, October 21, 1991, pp. 80-82.

The term multimedia has been used to describe everything from colossal extravaganzas to trade show slide presentations and is not exactly new. What is new is the price tag: hardware and software are now reasonable in cost, and the text and image editing programs require "no special expertise to master." Industry pioneers like MacroMind, Paracomp, Pixar and Adobe Systems are all in the battle for a piece of the pie. The results have been impressive, promising to take video production into the mainstream.

Slowing down multimedia development is the fact that there really isn't a broad consumer market at this time. Another problem is in educating the public: people need to realize that this is an entirely different medium from slide shows and video.

Multimedia cont.

Other issues relate to the use of archival images. Understandably archives are reluctant to grant use to images for this use as they would be copied over and over again.

Multimedia today occupies a shifting ground where the technological innovations and competing formats have combined to make an unpredictable outcome for the future. Electronic scanners, digital cameras, and other futuristic hardware are in the hands of a new breed of designers, where ideas are visual and communicated graphically and effectively.

Reference: "Breaking into Pictures" from HQW, by Ken Coupland, July/August 1991, Vol VI, No. 5, pp. 62-69.

Products boast of being multimedia or hyper-something, without a lot of comprehension among the general public about what is being talked about. In this article, "'multimedia authoring environment' means software that allows you to integrate sound, text, graphics, and animation or video into a user-controlled product." (p. 195)

A presentation could include animated charts, music from a CD ROM player, and video clips of demonstrations from a videodisc player with the user choosing what to view.

Reference: "MultiMedia Design Tools" by Nick West, MacWorld, November 1991, pp. 194-201.

With the Mac, the capability for producing multimedia presentations is clearly available, however, the software needed to fully and seamlessly integrate video on the Mac has appeared only in bits and pieces. Many of the complaints surrounding multimedia stem from unrealistic expectations.

Multimedia applications, especially the interactive ones, can't yet be based on a large body of experience from books, specialists, and training about interactive multimedia. Few Macintosh dealers are knowledgeable about video. Similarly few video dealers, professional or otherwise, are familiar with computers. Presently there is not a lot of help available, and developers are pretty much on their own to find information.

Reference: "Multimedia and Video" by Lon McQuillin, MacUser, Feb. 1991, pp. 4-32 in Buyer's Guide Supplement.

Bill Gates, 35 yr. old chairman of Microsoft Corp, who developed the basic software for the IBM PC may be the most powerful person in the PC business. Microsoft Windows software which makes all PCs as easy to use as Apple's Macintosh has become the industry's best-seller. Gates was in NYC to help launch the Multimedia PC Marketing Council, a group of companies dedicated to marketing computers that can run video, sound and compact discs simultaneously. According to Gates, the big problem in multimedia has been getting content into digital form so that it can be usable. People who have the expertise do not have the funding or knowledge to do this.

Reference: "Gates' Vision: Putting archives to use" by Bill Gates, USA Today, October 9, 1991, p. 8B.

Our culture has failed to deal with understanding the impact of the new technologies on society. Because understanding these dynamics is such a crucial element in designing, it is amazing that there has been such a complete failure to address these issues.

“If a quantitative change in some system environment changes past a certain threshold, you get an immediate qualitative change—very much a qualitative change. You change the scale of an airplane and suddenly it won’t fly—things of that sort.” quoted from Doug Engelbart 1987 (p. 24)

Reference: “The Augmentation System Framework” from Interactive Multimedia, by Sueann Ambron & Kristina Hooper, Microsoft Press, 1988.

Multimedia Software

There are 2 common approaches to creating multimedia presentations. One is to use an authoring system as its base. An authoring system (HyperCard, SuperCard, Plus, or Authorware) is ideal for specifying relationships among segments or user-controlled events. The other approach is to use an animation package as the starting point. Creating animation can stretch the capability of the computer to its limit, and several programs may have to be used to get the desired result. Where the authoring packages now support animation, animation programs now include authoring capabilities. (p. 52) So the line between authoring and animation which started out blurry has become even more so. Modeling software tags along with animation because it makes animation look so real. (p. 53)

Reference: “Authoring, Modeling, and Animation” by Lon McQuillin, MacUser, Feb. 1991, pp. 52-68 in Buyer’s Guide Supplement.

There are 5 representative approaches to multimedia software:

- Multimedia macro recorders: all screen activity is recorded into a file that can be played back for future use, also allowing sound to be added and sequenced. This is great for training applications, but limited in scope. Examples are MediaTracks and ScreenRecorder from Farallon Computing.
- Hypermedia programs: Hypermedia authoring applications, such as HyperCard, SuperCard, Authorware, Ask*Me 2000, Guide, Plus, or Toolbook, allow for visual effects, sound, and intelligent data manipulation. Their main strength is the underlying programming language, but animation is limited. They work best for electronic publishing applications. (p. 562)
- Animation software: Animator, FilmMaker, MacroMind Director, and Authorware allow for classic frame or 2-D animation. They are great for simple sequences, flying logos, and corporate identity advertising.
- Assembly or integration software: These programs take elements created by other programs and sequence them into full multimedia presentations. Examples are MacroMind Director, MediaMaker, and Ask*Me 2000. (p. 562)
- Modeling and rendering programs: Programs that create 3-D models that can be rendered and animated are being used to do limited animation work and also used as prototyping stations for larger computers. Examples include DynaPerspective, StrataVision 3d, RenderMan, Super 3D, and Swivel 3D. (p. 563)

Reference: Beyond the Desktop: Tools and Technology for Computer Publishing, by Barrie Sosinsky, New York: Bantam Books, pp. 549 - 597.

Multimedia Software cont.

In August of 1987 Apple's HyperCard was introduced and it is still the most widely used program. This program was provided for every Mac owner, and HyperTalk emerged as the model for multimedia scripting.

The first challengers of HyperCard were Spinnaker Plus (\$495) and Aldus SuperCard (\$299), both imitating HyperCard's interface. These three programs have become the workhorses of multimedia due to wide availability, low cost, a clear metaphor, and are ideal for creating simple projects quickly.

Anything more complex than card flipping requires using the scripting languages, which, once learned, are powerful tools. XCMDs allow control of external devices, such as videodiscs and CD ROM players. HyperCard's popularity comes not only from being bundled with the Mac, but also the ease of its use has made it a recognized tool and "spawned proliferating links to other multimedia authoring tools." (p. 196) Unfortunately, the new version has not made the use of color any easier for HyperCard users, unlike Plus and SuperCard.

Reference: "MultiMedia Design Tools" by Nick West, MacWorld, November 1991, pp. 194-201.

HyperCard is the first of a new generation of environments of programming optimized for easy scripting by non-specialists and simultaneous work with text, image, and sound. (p xiii)

It is known for the creation of highly interactive multimedia programs supporting creation of multiple branching, free-search environments. It has commands that allow easy linkage of text, image, and sound and also built-in commands and structures to allow the program to monitor the user's actions with the mouse and keyboard in order for the program to respond. (p. 4)

Its programming language, HyperTalk, is very close to regular English, (p. 6) and in addition, it makes use of metaphors that link its language elements to ordinary objects such as buttons and cards. (p. 7)

Core innovations of HyperCard are its use of an everyday programming language, its use of modular objects, the ease of incorporating multimedia, as well as its capabilities to provide for interactivity. (390) "There is a growing realization that different kinds of information can be conveyed by image and sound that are not easily communicated by text, and consequently there is a push to expand everyday communication media." (392) In a HyperCard environment, images and sound become as easy to manipulate as text. Building into presentations the possibility of interactivity of the user allows the program to read the changing desires and associations of the viewer. This means designers need to consider their audience more carefully than ever. (P. 389-392)

Reference: MultiMedia Design with HyperCard by Stephen Wilson, Englewood Cliffs, NJ: Prentice Hall, 1991.

New features were aimed at Hypertalk programmers:
the powerful new script editor
the built-in debugger
the new Hypertalk commands

Consider customizing your home stack to open various palettes and tools at startup by adding the commands to the script listed below.

The Palette Maker in the tools stack allows you to create customized palettes.

Commands for Home Stack

—added for custom window display on 13" monitor

show card window at 8, 46

show pattern window at 520, -8

show tool window at 520, 136

palette "navigator", "520,280"

show scroll window at 0, 357

show message at 102, 388

—end of added statements

Reference: "Insights on HyperCard 2" by George Beekman, MacWorld, September 1991, pp. 297-300.

Spinnaker Plus (\$495) has full color features and scripts can be attached to any object (incl. graphics), but it does not allow more than one stack to be open at a time. You can choose one of eleven different palettes or create your own. Because of the integral use of color, it runs more slowly than black and white HyperCard, but about the same speed as HyperCard using color elements.

SuperCard, (\$299) the most elaborate of this style of authoring program, has its basic limitation in the need for an authoring module (SuperEdit) and a run-time module (SuperCard). Moving between the two can be an annoyance, and in addition, SuperCard is the slowest of the three.

SuperCard supports PICS animation files and has its own accelerated animation format, called STEP. Of any HyperCard-style program, it has the most extensive set of tools for building projects. For complex projects in color and control over multiple windows, SuperCard is a good choice.

Authorware Professional (\$8000; \$995 for educational institutions) is geared for the educational and training market. Authorware provides strong features for tracking students' progress with superb features for student testing. In addition, purchase of the program includes a session at a training center and extensive on-line support.

Authorware uses an icon-based flowchart metaphor called the flow line. The space where the program is designed doubles as an easy-to-follow map, perhaps "the most sophisticated visual aid on the market." (p. 197) It has built in controls for external devices and is moderately easy to learn and extremely powerful. The cost makes it prohibitive.

Multimedia Software cont.

Course Builder (\$1495) resembles Authorware in its emphasis on educational uses, the control over tracking and responding to student input. The main metaphor here is the course map created by choosing from various states (graphics, video, etc.) and connects with routes on the map. This metaphor does not always organize neatly. Because Course Builder avoids a scripting language (all choices are available in dialog boxes), it is sometimes limiting. In addition it does not integrate video and animation smoothly.

Director (\$995) contains tools for creating high end animations and presentations with varying degrees of interactivity from the simple to complex. It contains two modules: the Overview module allows users to sequence multimedia objects created elsewhere with transitions and delays in between. The Studio is built on a film metaphor, and is difficult to learn. However the results can be dazzling and the effort is worth it.

The program provides relatively easy creations of animations, moving up to 24 objects at a time. This program uses Lingo as its scripting language (similar to HyperTalk). Scripts can be attached to any object or frame.

Reference: "MultiMedia Design Tools" by Nick West, MacWorld, November 1991, pp. 194-201.

Interactive MacroMind documents can be created without high costs of multimedia scripting, story boarding, production and post-production processes. Creative control, quick turnaround, and cost-effectiveness can be expected.

The most popular interactive application is the instructional presentation or orientation. Moving graphs, animated graphics, sounds, and music can turn a dull briefing into an experience that people remember and enjoy. Learning is always more fun when you can interact freely with the instructor, and the goal of interactivity is to allow that interaction to happen. Simulations, demonstrations, and visualizations are at their most effective when the viewer or presenter has control over the information flow.

Reference: "MacroMind Director: Interactivity Manual" by MacroMind, Inc., April 1990, pp. 4-5.

Linx provides several interesting presentation tools with tools for archiving. Also it allows previously produced multimedia materials to be linked, instead of pouring all the resources into one gigantic program. This is especially helpful in the case where materials need to be updated frequently. In addition Linx can control up to six monitors simultaneously and can split each of these monitors into nine zones, each zone displaying a separate graphic or animation.

There are some rough edges here though. All the applications files must be placed on the desk top; many paint programs wipe out the link; and finally there is a steep cost (\$9000).

The increasing popularity of multimedia can be measured by the many programs that contain multimedia elements but are not full-fledged authoring tools:

OWL International's Guide 2.0 (\$295) a text specialist

Opcode Systems' Max (\$399.95) MIDI control for musicians

Farallon's MediaTracks (\$295) designed for computer training

Paracomp's FilmMaker (\$695) specialized animation tool

MacroMinds MediaMaker (\$695) tools for converting multimedia presentations to videotape.

Whitney Educational Services' VideoDiscWriter (\$389) a HyperCard add-on, helps create programs with VideoDiscs

HyperPro's VideoAuthor (\$385) organizes videodisc segments

Reference: "MultiMedia Design Tools" by Nick West, MacWorld, November 1991, pp. 194-201.

"The kinds of multimedia projects that you can undertake are largely predicted by the capabilities of the computer equipment. Not surprisingly, therefore, manufacturers stress different approaches (p. 568) to multimedia that paint their computers in a favorable light. The three major PC platforms for multimedia are the Macintosh, IBM PC, and Commodore Amiga." (p. 569)

"Multimedia is very equipment intensive, and major differences are apparent when we examine the high-performance graphics necessary for animation, high-quality audio, and video. Some of these differences are radical and underpinned by the design philosophy of each manufacturer." (p. 569)

"Apple regards multimedia as a natural extension of desktop publishing and refers to it as desktop media, including in that definition the entertainment, training and presentation markets." Also the authoring software is a particular strength of the Mac. "HyperCard is the first step on the road to what John Scully (Apple's CEO) calls 'the Knowledge Navigator.' The Navigator concept envisions a portable computer with a multimedia database incorporating artificial intelligence. Agents built into the software would search for, store, and filter information, using past work as a guideline." (p. 569) Other advantages of the Mac are its high-quality color display and output, supports 8-channel stereo with MIDI capabilities. (p.570)

"Although IBM PCs have strong processing power and good new video standards. . . , it is the least capable multimedia personal computer right out of the box. However it has the best third-party add-on support and is the biggest segment of the multimedia market. Windows and OS/2 offer better memory management, multitasking, standard file formats, and device support and are more actively supporting multimedia developers than MS-DOS. However, multimedia on the PC is a more complicated and expensive proposition than on other personal computers due to a lesser degree of standardization." (p. 570) The PC has in it one channel for sound, other boards can be added; software lags behind the Mac, and animation programs look primitive. (p. 571)

Multimedia Hardware

Multimedia Hardware cont.

The Amiga, introduced in 1985, pioneered desktop video, with its success largely attributable to its impact on the desktop video market. "In addition to the multitasking operating system, the Amiga ships with a set of coprocessor chips specifically designed to speed up graphics, video, and I/O." (p. 571) The platform is blessed with good low-cost equipment, mature authoring software, and vigorous third-party market.

Amiga was designed with most multimedia support right in the hardware. Only the area on the screen that changes is updated increasing screen performance. This system is designed to switch between interlaced, noninterlaced, and overscan video modes. (p. 572) Its cost, operating system and sophistication make it the choice of many video professionals. (p. 573)

Reference: Beyond the Desktop: Tools and Technology for Computer Publishing, by Barrie Sosinsky, New York: Bantam Books, pp. 549 - 597.

IBM has unveiled its first platform specifically designed for multimedia, the PS/2 Ultimedia Model M57 SLC. Priced beginning at \$6000, it has built-in multimedia features including a CD-ROM drive that's also compatible with CD audio, speaker and headphones, and high-resolution XGA graphics.

This operates on any of three operating systems - DOS 5.0, Microsoft Windows 3.0 with Multimedia Extensions, and OS/2 version 2.0 (to be available this March 92). All three of these operating systems plus a demo come on a CD-ROM disc packaged with the M57.

A number of products are being designed to be released at the same time: Columbus: Encounter, Discovery and Beyond; and The Illuminated Books and Manuscripts.

Reference: "Columbus, New Products, Fill IBM Sails" by Jim Strothman in Computer Pictures, Oct.-Nov. 1991, pp. 64-65.

Multimedia has come to the PC with the introduction of Multimedia Windows. Besides a 286 10-MHz system with VGA graphics, 2MB RAM, a 30 MB hard drive and a 1.44MB 3-1/2 in floppy drive, you'll also need a CD-ROM player with 150K/sec transfer rate. Microsoft has lined up some of the biggest names in software to help complete the hardware systems for buyers. In addition to Multimedia Windows, there is ToolBook, Authorware Professional, Guide, IconAuthor, MediaBase, and Windowcraft; all authoring programs that allow nonprogrammers to create multimedia presentations.

Will it succeed? Slowly at first, but as the prices come down, expect to see a lot more done on the PC in multimedia.

Reference: "Make Way for Multimedia" by David English, Compute, June 1991, pp. 20-25.

While multimedia technology is ideal for presentation, presentations are often given to groups too large to cluster around an SE's monitor. The three most common displays for group presentations are large-screen CRT monitors, video projectors, and LCD overhead-projection panels.

Direct-View Monitors. The largest monitors currently available are the 37-inch units, generally priced around \$6,000; these come with RGB inputs, have multiple scan rates, and provide a large crisp display. Their size and weight (100 lbs) make them suitable for part of a permanent display.

Video Projectors. These come in several forms. One of the most established types uses three projection tubes driven by specialized CRT screens encased in mirrored enclosures. These use the same technology as most home projection systems, but are front-projection designs instead of rear-projection. Barco, Electrohome, Panasonic, and Sony are examples. The other type is the GE light-valve projector, which uses a projection bulb to produce a brightness level sufficient for large images and use in theaters or auditoriums.

LCD Overhead Projector Panels. Relatively new, color has been introduced, and the sale of these has exploded. Capable of producing 17 hues, it can match the Mac screen's colors, but not the intensity and runs about \$6000.

Reference: "Multimedia and Video" by Lon McQuillin, MacUser, Feb. 1991, pp. 4-32 in Buyer's Guide Supplement.

Graphic Design is a profession that has been emerging over the last 60 years; it is a combination of the disciplines of photography, printing and typography, art, and advertising. (p. xi)

Reference: Nine Pioneers in American Graphic Design, by R. Roger Remington and Barbara J. Hodik, Cambridge: MIT Press, 1989.

Color in presentations not only make them more impressive, but also more understandable. Color delineates the elements more sharply than the obscure and cluttered hatch patterns or black and white. Color effectively increases the information density and clarity of the images, or in other words, it lets you say more with less.

Packages today for presentation graphics are extremely powerful, but increased artistic power doesn't necessarily mean increased artistic talent. Making presentations unique requires attention-getting images, and for the non artist, that's where clip art comes in. Another way to get eye catching images is to scan them. Presently a 24-bit scanner with 300 dpi resolution is available for less than \$3000.

Reference: "Power Presentations: A New, Colorful Era" by Rock Miller in PC Magazine, May 14, 1991, (Vol. 10, No. 9), pp. 101-104, 192.

It is a well known fact that audiences can easily become bored. Being a successful presenter means being able to edit, structure and illustrate your ideas to inform, influence, and entertain the audience.

Presentation Graphics

Presentation Graphics cont.

Because larger typefaces have greater visual impact, messages must be edited to a few very meaningful words that convey the meaning. Consistency is a virtue, but so is knowing when to break the rules. Progressive builds can create a pseudo-animated effect. Charts can convey multiple concepts in a well-done drawing, and many presentation programs have limited spread sheet capabilities built in.

Reference: "Presentation Graphics" by Luisa Simone in PC Magazine, May 14, 1991, (Vol. 10, No. 9), pp. 107-192.

Advice on creating stacks:

1. Know your audience.
2. Develop clear and simple organization.
3. Beyond consistent button placement and understandable icons, stay one step ahead of users by guiding and anticipating user actions.
4. "Just as with books and magazines, the graphic design of hypermedia is a powerful factor in its ability to communicate. The look of a stack or project defines its personality, helps state its purpose, and directs the user to important information." (p. 127)
5. Skillfully integrate text, graphics, sound, and animation.

Reference: "The Secret of the SuperStacks" by Liza Weiman, MacWorld, April 1991, pp. 124-129.

Following are several tips for presenters to keep in mind.

1. Keep the media in mind as the presentation is designed. An overhead transparency requires a different set of design and layout rules than an on-screen slide show.
2. The purpose of the presentation is to effectively reinforce the speaker's message, not to make things look pretty.
3. Use special effects to your advantage. They should point to the message and reinforce it, not detract from it.
4. Consider using layers or builds to add a sense of motion and interest, and also focus the audience's attention.
5. Use colors of high contrast and different intensities for best results.
6. Irregular graphics are more interesting and exciting; squares, circles can be more calming.

Reference: "Presentations" by Keith Thompson in PC Publishing and Presentations, Aug-Sept. 1991, pp. 24-26.

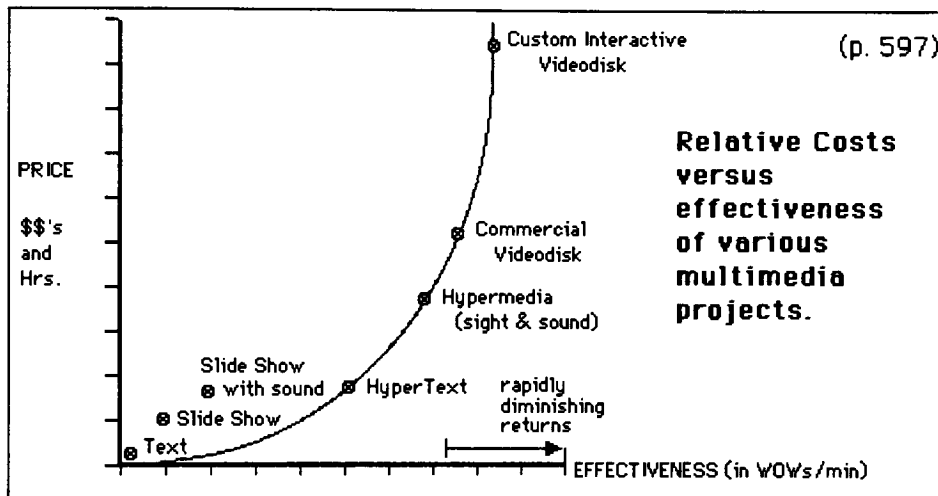
Using type on-screen involves special considerations.

1. For on-screen presentations, use a type manager utility, then you don't have to worry that a 26 point size will look bad.
2. Keep text on slides to a minimum. Use a maximum of 6 lines of 6-7 words each. For more text than that, use another slide.
3. Larger type is more noticeable than smaller type.
4. Use bold type faces instead of underlining (a holdover from days when typewriters couldn't do bold).
5. Avoid italics as the small lines tend to get lost when projected. Shadowed, outline and fancy fonts should be avoided.
6. Avoid the use of all CAPS as they are harder to read.
7. Limit the number of fonts per slide (usually a sans serif for headlines and a serif for the text).

Reference: "Presentations" by Keith Thompson in PC Publishing and Presentations, Aug-Sept. 1991, pp. 24-26.

Artists are beginning to explore the possibilities of works weaving sound, images and text into new multimedia tapestries. HyperCard can interactively control external devices such as videodiscs, MIDI sound synthesizers, and kinetic environments. Examples include "We Make Memories" by Abbe Don (a family scrapbook approach), "HyperPoems" by William Dickey (nonlinear poetry & imagery), "Eat" by Michael Naimark (an installation about consumption), "The Manhole" by Robyn & Rand Miller (interactive fantasy world), "Deep Contact" by Lynn Hershman & Sara Roberts (a mystery), "Xherone 1.3" (an idiosyncratic sequencer of image & sound), & "Father Why" by Stephen Wilson (controls speech, music & lights).(p. 7-16)

Reference: MultiMedia Design with HyperCard by Stephen Wilson, Englewood Cliffs, NJ: Prentice Hall, 1991.



Evaluating Systems

A simple rule for evaluating the quality of a particular computer or authoring system is to examine the best games currently available. Games tax the limits of software in the quality of graphics, sound, and animation that they use. "Games are written to get the biggest bang for the buck, and are among the most successful software programs on the market today." (p. 567)

Some games to learn from include Colony, written by David Alan Smith, a researcher in virtual reality programming, as an exercise to determine what is possible on a personal computer, Bomber, a HyperCard game; also Manhole, and Cosmic Osmo, both published by Activision, are HyperCard stacks. (p. 568)

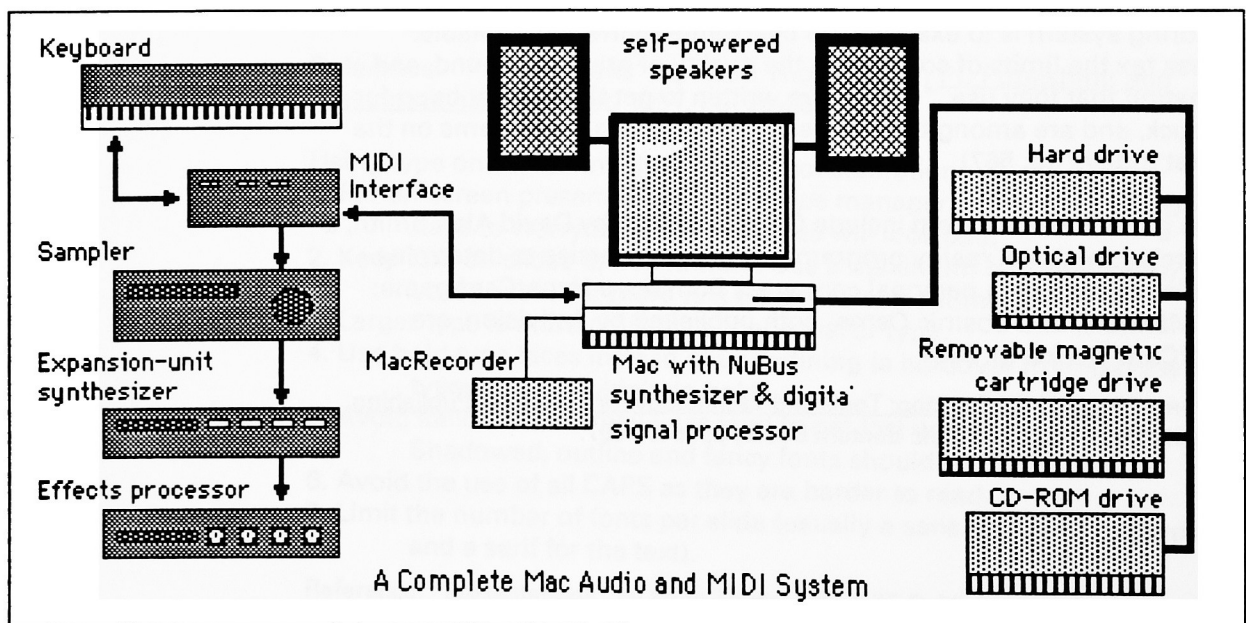
Reference: Beyond the Desktop: Tools and Technology for Computer Publishing. by Barrie Sosinsky, New York: Bantam Books, pp. 549 - 597.

Audio

The importance of sound was recognized by the creators of the first Mac. Sound can be added to Mac multimedia applications more easily and less expensively than video, animation, or any other enhancement. Stereo sound ports come standard with every Mac except the LC. Sound input ports and microphones come standard with the LC and IIsx, and third party products allow sound to be incorporated to the other Macs. Having so much sound capability standard on the Mac encourages more sound products to be developed. (p. 38) Most computer programs have not made use of sound effectively to date, and the question arises, is it necessary. Once sound was added to movies, how many silent films are now being made? The once exclusive music-studio sound can now be included in multimedia. (p. 48)

In addition to generating sounds the Mac has become the musician's instrument of choice for controlling MIDI (musical-instrument-digital interface) devices. In fact it has become such a standard "that even mammoth Synclavier, a leading synthesizer vendor for the pros, abandoned its proprietary interface for that of the Mac. (p. 8) Its industry standard protocol gives Mac "the ability to conduct virtual symphony orchestras." (p. 9)

"MIDI devices such as keyboards communicate with the Mac via a MIDI interface connected to the Mac's serial port(s). Additional MIDI devices such as samplers, expansion-unit synthesizers, and effects processors can be driven via a MIDI loop-through chain. The Mac can be equipped with a NuBus synthesizer and a digital-signal processor that lets it record digital audio directly onto hard disks. With the MacRecorder, the sound can be recorded on standard Macs, and self-powered speakers give the Mac a more powerful sound. Audio recording requires prodigious disk capacities, leading to the need for large hard disks, optical drives, and 44-megabyte cartridge drives for transporting files. CD-ROM drives can be used to load sampler data files from CD-ROM discs and digital audio from CD libraries." (p. 38)



MIDI (musical-instrument-digital interface) is a type of local-area-network protocol for sound, providing communication paths among instruments, synthesizers, computers, and playback equipment. (see diagram prev) MIDI is made possible when synthesizers became digitally controllable devices. Actually a synthesizer is a dedicated computer with a piano keyboard instead of an alphanumeric keyboard, and sometimes it has both.

It is now possible to do anything that used to require a recording studio, from overdubbing to film scoring at home with everything linked to a Mac through a MIDI interface that costs less than \$100. MIDI specifications provide for up to 16 channels of communication among MIDI devices, which can be increased with other interfaces. "Sampling is the digitization of analog music sources by sampling the source at frequent enough intervals to (hopefully) fool the human ear." (p. 40) A typical rate of sampling the analog signal would be 44,000 or so times a second. This takes a lot of computer power.

The MacRecorder is probably the most familiar sound application to Mac users. Sound can easily be digitized from a number of sources and incorporated into an application or stack. This is probably the "source of all those funny sounds you've no doubt heard coming from your hacker friends' machines."

But to stop there is to sell the MacRecorder short; the MacRecorder has enough sophistication to make it valuable in professional productions as well. HyperCard, SuperCard, and MacroMind Director can all play back sound resources, such as those created with MacRecorder.

For higher sound recreation, two methods are available. Audiomedia (a Digidesign board) can play back files from within HyperCard, SuperCard, or MacroMind Director, using Sound Access, and XCMD/XFCN provided with Audiomedia. Audiomedia parallels the MacRecorder but produces a substantially higher level of sound quality. (\$995) MIDI playback from within HyperCard, SuperCard, or MacroMind Director is possible with MIDIplay from Opcode (\$60) or HyperMusic from Passport. (\$80) (p. 44)

Digital audio places great demands on disk space. When you use Sound Tools, a 60-min. recording can fill a 600 meg hard disk. That is why the removable hard-disk cartridges are so popular among musicians. Street prices have dropped to less than \$700 for the drives and to less than \$90 for the cartridges.

For larger needs, reusable optical discs would prove attractive if access times could be improved substantially. An erasable optical cartridge can hold about 300 meg per side, accommodating around 28 minutes of digitized audio. CD-ROM is also promising; Discs are becoming increasingly easier to publish, and list prices which hover around \$800 are expected to drop. There are a variety of disc formats to store different types of information for different media. (p46,48)

Reference: "Multimedia and Audio" by Ken Gruberman and Lon McQuillin, MacUser, Feb. 1991, pp. 38-50 in Buyer's Guide Supplement.

Audio cont.

CDs were designed to store music, not pictures or computer information, and their data retrieval rates are limited. There is often an annoying pause while the CD drive fetches a new screenful of information - giving the machine a sluggish quality that people used to the furious pace of TV shows and video games may deem unacceptable. They are just too slow.

More difficult than that is the fact that most of the competing devices are incompatible. With the exception of the MPC (a personal computer with the CD drive built in) a disc from one company will not necessarily work on another company's machine. This breeds confusion and consumer resistance. However in the industry there is optimism that these problems will be worked out in the coming years.

Reference: "The World on a Screen" by Philip Elmer-Dewitt, Time, October 21, 1991, pp. 80-82.

Current applications include Warner New Media's "The Magic Flute Audio Notes" based on Mozart's "The Magic Flute." The HyperCard stack has 7000 screens and is interactive in two languages. The next planned release (Beethoven • String Quartet No. 14) does a better job. Unfortunately, the product's value for those trying to learn how to create interactive products is limited as the stacks are locked. Voyager's products are open however, and increases their value in this regard. Voyager has three notable products: The Voyager CD AudioStack, which enables HyperCard users to work with software tools to control CD-ROM discs, Ludwig van Beethoven - Symphony No. 9, an excellent model of interactive design, providing many choices for the user, and Igor Stravinsky, the next release planned. (p.48)

Reference: "Multimedia and Audio" by Ken Gruberman and Lon McQuillin, MacUser, Feb. 1991, pp. 38-50 in Buyer's Guide Supplement.

The original Mac included a sophisticated electronic sound-generating integrated circuit chip and speaker as part of the basic hardware. Newer models provide even more sophisticated sound possibilities.

HyperCard allows for digitally recorded sounds that can be used in conjunction with sound digitizers such as the MacRecorder. External commands allow scripts to talk with a synthesized voice or to control MIDI-compatible music synthesizers. The result is that multimedia events can integrate image with music, speech and sound effects. (p. 211)

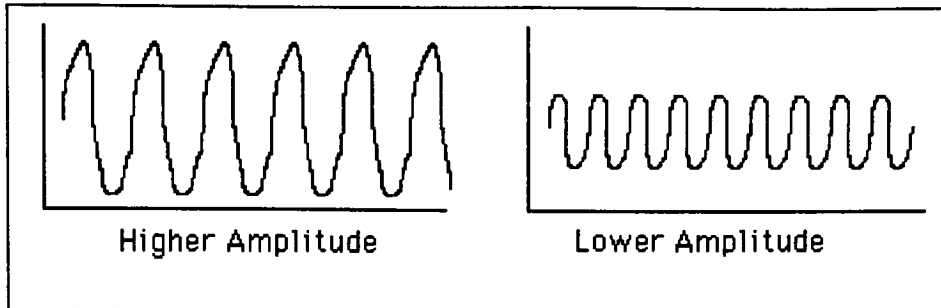
Understanding Sound

For humans, sound is experienced as the air vibrates against the eardrum and connected sensory organs of the ear. In the computer, the sound originates by the electronic control of the diaphragm of a speaker, which vibrates the air, and ultimately the eardrum.

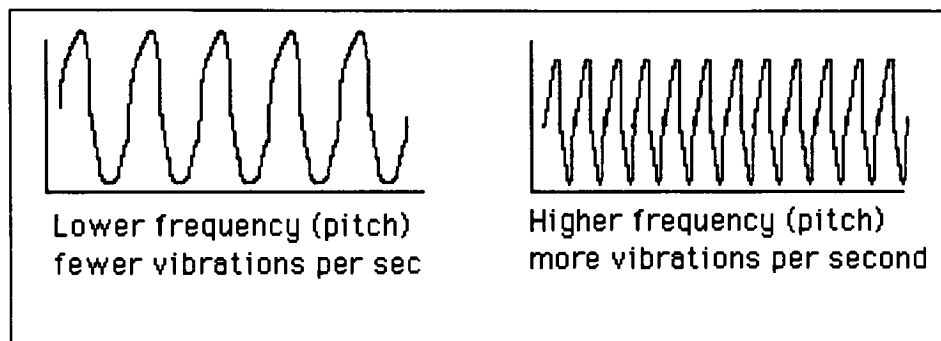
Sound is characterized by certain analytical features, which give the artist discrete control over their values. These include its amplitude, or loudness, the frequency or pitch, (p. 212) and the overtones or harmonics. (p. 213)

The amount of energy used in creating a sound determines its amplitude, or loudness. A great deal of movement produces a louder sound. (p. 212)

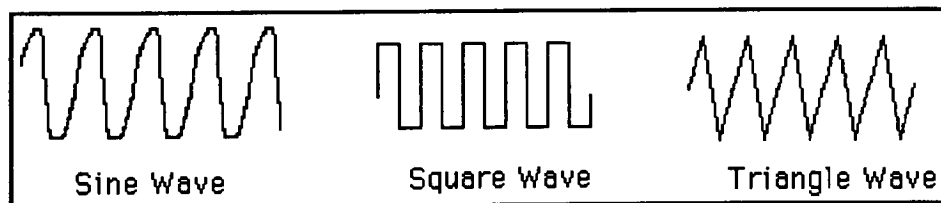
Understanding Sound cont.



The speed with which the vibrations occur is called the frequency or pitch (highness or lowness of a sound) measured in hertz (hz) which means vibrations per second. Human hearing can pick up between 20 and 20,000 hz with middle C at 261.6 hz. (p. 212)



Obviously the same note can be sounded by different instruments or voices and does not sound the same. The differences in sound are the result of variations in the frequency of the vibrations called overtones or harmonics. The variations in the wave shapes occur in basically three common shapes:



Each vibration can build up slowly and go slowly or come and go almost instantaneously. (p. 213)

Also the way a sound ebbs and flows in volume over time differs. The shape of this variation is called the ADSR envelope (stands for attack, decay, sustain, and release). A trumpet sound is a short blast, while a harp sound dies out slowly making the envelopes quite different. This combination of features is called the timbre or color of the sound.

Synthesized Sound

Sophisticated computer chip-activated sound synthesizers allow artists and musicians to manipulate each of these qualities of sound. The internal chip in the Mac does not offer such a level of discrete control, it does offer some control. (p. 214)

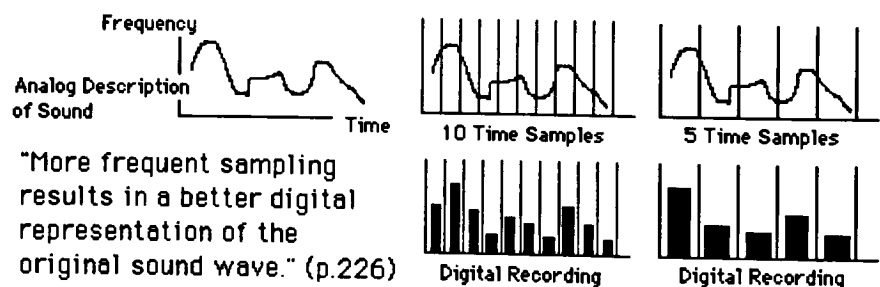
The Play command in HyperCard is the workhorse for sound generation. It allows for the playing of a variety of sounds at a variety of tones in various tempos (speeds). "The sound parameter specifies a particular sound definition (sound wave & color) that will be used to create the sound generated by the play command." (p. 214) There are 4 built in sounds to HyperCard: harpsichord, boing, silence, and dialing tones. This repertoire can be increased using a sound digitizer.

HyperCard also has a sound function (the sound) that will return the name of the sound playing or the word "done" if the sound is finished. Checking the status of a sound allows for more precise sequencing, meaning that in moving from one card to another a sound is not interrupted to start another or eliminating repeated calls to play a sound that could crash the Mac. (p. 214-218)

Sound can be coordinated within scripts to certain actions by using variables. As an object moves up, for example, the pitch increases. It can also be coordinated to certain events occurring on the screen either to draw attention or to complement an action. (pp. 219-220)

Many computer chips are capable of synthesizing human speech using sounds to approximate the phonemes (basic sound building blocks of speech). Often these are robot-like and difficult to understand. Using XFCNs and XCMDs MacInTalk can be used in HyperCard. It can be somewhat tricky because there are approximately 60 basic phonemes used and words need to be spelled phonetically to sound right. (p. 221-223)

Digital or sampled sound is a radically different way to generate sound than the synthesized sound method. In the synthesized method, the software mathematically creates waveforms that instruct the sound chip to make sounds.



Digitized sound methods record real sounds in a form that can be used to control the sound chip. To accomplish this, the sound signal must be converted using special software and hardware into a digital form that the computer can work with. Sound, like light, is analog (sound is a continuous variation of loudness and frequency). Because computers work digitally, sound must be converted from analog to digital for the computer to make use of the information. This is accomplished by a process called sampling. (pp. 223-225)

Unlike synthesis, digitization preserves much of the nuance and color of the original. HyperCard can play any digitized sound that has been recorded in the appropriate format, so that stacks can incorporate relatively high-quality sounds. (p. 226)

Digital sound recording and manipulation is made possible by a specialized sound-editing program called SoundEdit. If a microphone is connected to the sound jack, then SoundEdit can record any sound, play it back, and allow for its manipulation. Sound can be edited much like text, with cut and paste capabilities or transformed in many ways. "One has the equivalent of a sound studio that a few years ago would have cost tens of thousands of dollars." (p. 231) Tools are: amplify (loudness/softness), backwards (reverse sound), bender (changes frequency), echo, envelope (adds pattern of changing loudness), filter (differentially controlled frequency ranges), flanger (adds pitch to unpitched sounds), etc. (pp. 228-234)

Reference: MultiMedia Design with HyperCard by Stephen Wilson, Englewood Cliffs, NJ: Prentice Hall, 1991.

The Musical Instrument Digital Interface (MIDI) is a universal standard created in 1929. It specifies protocols for wiring and connectors that enable computers and instruments to communicate. MIDI would allow a network of instruments that could be played from a single device, would let a computer record and play back music, or in essence, create a music studio. (p. 579) MIDI allows up to 16 channels of simultaneous instruments, and communication at 31.25 Kbits/sec, which is fast enough so that a chord does not sound like its separate components. (p. 580) HookUp! (object oriented control software) allows devices to be linked by drawing connections in order to create multimedia sequences, to help beginners in hookup of MIDI. (p. 581)

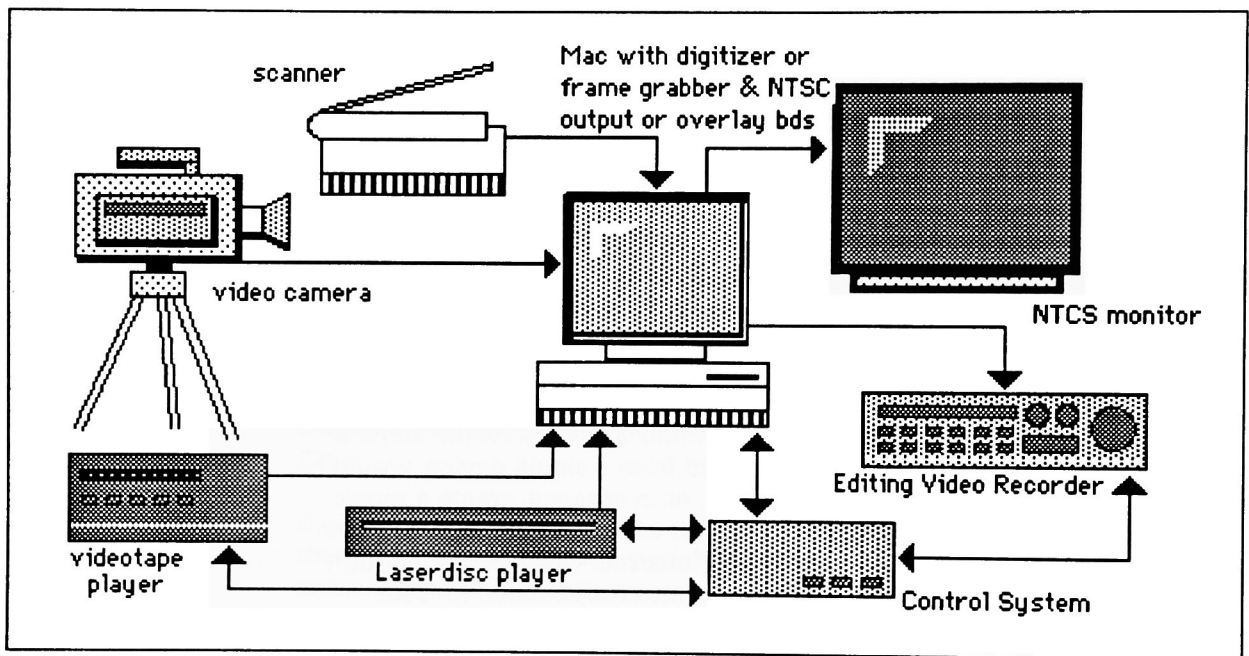
MIDI Interface

Sequencers are programs that let you record and play back MIDI data. With such a program, a musical piece can be written for up to 16 instruments. The sequence can be edited and played back just as if your computer were an elaborate digital musical instrument. Patch editors can store thousands of different sounds and settings for a synthesizer. It might take 100 Mbytes to store ten minutes of digital stereo sound, but a ten minute MIDI sequence uses only 30 Kbytes or so. Tip for multimedia projects: Whenever possible use MIDI commands and a sequencer in place of sampled recorded sound. Music recorded in this way has greatly compacted sound files and quality at the level of your playback devices. (p. 581)

A scoring program enables you to write and print music using conventional notation (The Adobe Sonata font is used in most of these programs.) A scoring program is to composers what a word processor is to writers. Many scoring programs actually write the notation as the instruments are played, although some cleanup is required due to slightly misplaced notes. Scoring in this way has been the fantasy of most musicians; programs include Professional Composer, NoteWriter and Encore, and Deluxe Music Construction Set.

New software promises to create the ultimate work station: a professional-quality home recording studio that can do music editing, audio postproduction, and produce master recordings. (p. 582)

Reference: Beyond the Desktop: Tools and Technology for Computer Publishing. by Barrie Sosinsky, New York: Bantam Books, pp. 549 - 597.



Video

Adding video capability to your system can transform presentations, simulations, etc. A NuBus board that provides out put to a VCR costs about \$600, although high end equip can be as much as \$10,000. Still video frames can be captured, live video can be displayed from external source, Mac-created graphics can be recorded on videotape or transfer by 3M or Pioneer can be done to a laserdisc for interactive applications. Peripheral devices such as VCRs and laserdisc players can be controlled. Best of all these functions can be combined. (p. 579)

To move images between the Mac and video equipment, there are two conversion problems when you use a Mac with standard video equipment. First, analog images must be converted into digital ones (called A-to-D or A/D converting) in importing video into the Mac environment, and then digital ones must be converted to analog (called D-to-A or D/A converting) when displaying or recording Mac images on standard video equipment. Second, the scanning system used to create the Mac's display differs from that used for NTSC video (the kind of video American TV uses). Standard Mac displays scan sequentially, in a noninterlaced mode, one line at a time, from the top to the bottom of the screen. NTSC video uses interlacing scanning images in two passes.

Even as the Mac community deals with NTSC video, the target is moving. High-definition TV (HDTV) will almost certainly supersede (sic) NTSC video at some point. HDTV will bring a better, clearer, and wider-screen video standard. It will double current NTSC resolution and increase the aspect ratio from 4:3 to 16:9. Some movies and commercials are already being made with HDTV equipment. Still, presently you can still see scan lines, and unfortunately, as with TV today, there probably won't be a single worldwide standard. The biggest challenge is the amount of data contained in each image (two to six times the spectrum space allotted today), so it will require advanced compression techniques.

The videodisc is an analog image-storage medium that uses technology similar to that of audio compact discs. There are two varieties: CAV (constant angular velocity) and CLV (constant linear velocity), each of which can be read by any videodisc player. CAV discs store one frame per revolution (this facilitates still-frame and slow-motion effects, and the process of finding particular frames rapidly). CLV records tracks of equal length contiguously, regardless of where they fall in the CLV's spiral track. The trade-off is that CAV discs limit you to half an hour per side, whereas CLV discs give you a full hour per side. The latest machines contain frame buffers that can provide for still-frame and slow-motion effects even with CLV.

In small quantities, videodiscs cost a lot to produce, but prices have been coming down in recent years. It is possible to have a single LaserVision-compatible disc, called a check disc, pressed for \$100 to \$125. A single distribution disc costs about \$360, and a thousand will run about \$10 each. 3M Optical Recording in St. Paul, MN is the premier company that offers videodisc- mastering and -replication services.

However the costs of mastering and replication pale next to the costs of designing and producing an interactive videodisc. Hiring a production company that specializes in creating interactive programs, a very simple, straightforward production of 15 to 20 minutes' duration will cost around \$50,000; a more sophisticated production will be from \$100,000 to \$250,000.

Reference: "Multimedia and Video" by Lon McQuillin, MacUser, Feb. 1991, pp. 4-32 in Buyer's Guide Supplement.

Video cont.

Among the best of the videodiscs being developed for sale is a series from ABC News InterActive that allows its users to explore subjects like the AIDS epidemic or the life of Martin Luther King Jr., by roaming through film and video clips culled from ABC's extensive library of news footage. In some cases, these clips are supplemented by printed matter, so that someone interested in King's "I have a dream" speech can not only see a film of the speech and read its text but can also call up background information on everything from the Voting Rights Act of 1965 to relevant Bible passages. Good interactive multimedia can be extremely costly to produce.

Reference: "The World on a Screen" by Philip Elmer-Dewitt, Time, October 21, 1991, pp. 80-82.

There are three important issues in computer video: design, storage, and delivery. Design is an authoring software problem; storage involves memory, compression, and file format issues, and delivery is a hardware problem that affects the choice of computer and peripherals.

Moving a video signal in and out of the computer in a format that a TV or video film can use involves 2 conversions: first a digital to analog (DAC) in the computer-to-analog RGB display must be made, then analog RGB must be encoded to NTSC (National Television System Committee), PAL (European Phase Alternating Line, or the alternate European encoder SECAM. RGB to NTSC conversion is called encoding; the reverse process is called decoding. (p. 583)

NTSC is an interlaced mode with a horizontal scan rate of 15,750 Hz (525 lines, 30 frames/sec) that is a 35 year old standard, modified in the 1950s to allow for compatible color.

Color was accomplished by adding an additional signal for chrominance (color) to the original luminance (brightness) used in black and white pictures. This standard is of lower image quality than computer video, and will probably be replaced by HDTV (high definition TV). (p. 583)

TIP: Monitor flicker is a significant problem: Avoid dithered patterns, single or odd-numbered pixel lines, bright oranges or reds, high contrasts (black next to white) and fonts smaller than 18 pt. Always use antialiasing to soften edges. Design for the center 80% of the screen. (p. 584-5)

Video digitizers produce images that can be saved in common graphics files such as PICT2 or TIFF for the Mac, or .TIF or .TGA for the PC. Video digitizers acquire images using a video camera, so the resolution of the images obtained is a constant set by the hardware. Focus and lighting are also important issues. Digitizers create images with a fixed array of CCD's (light sensitive transistor diodes) and are limited in resolution by the size of each CCD in the array. Scanners, by translating their array of CCDs across an object, minimize some of this limitation.

With digitizers, halftones must be produced at the input end, and are not of the same quality as those created by scanners. (p. 585)

Systems range from \$99 (DAK Industries) up to \$2500 for putting TV in a small window within the computer, for those who can't work at a computer without the TV. A notable product is RadiusTV, which displays live video or TV (RGB or NTSC) using a Mac. Images can be captured as PICT files, and sound can be digitized.

Beyond that a complete desktop video editing setup costs around \$10,000 and would include a computer with speed, power, and memory, storage devices (large and fast drives), Genlock/Encoder (needed to record to a VCR), VCRs (capable of genlocking, and with flying erase head), and input devices (such as camera or camcorder, lights, lenses, and other accessories.) (p. 588)

Videodiscs store sound, text, still pictures and full motion video with a 10-year lifetime. While videotape is more prevalent, it does not work well in multimedia applications due to its linear access. Lengthy searches for material destroy interactivity. (p. 588) In addition to fast random searches, videodisks also offer better picture quality, and are ideal for interactive applications. (p. 589) The 2 most common methods for recording are constant linear velocity (CLV) and constant angular velocity (CAV), both of which can be read by any videodisk player. CAV, preferred for multimedia, stores one frame per revolution and allows slow motion forward and backward, freeze frame, and rapid frame searches. (p. 590)

Reference: Beyond the Desktop: Tools and Technology for Computer Publishing. by Barrie Sosinsky, New York: Bantam Books, pp. 549 - 597.

The JASON project was an attempt to use state-of-the-art technology to bring video images in real time from the deep sea back to museums around North America. This project was carried out with a development team using HyperCard. Development took nearly a year with students researching and programming some parts of it. The parts were put together. In using it, the users were asked open ended questions to help in the evaluation process of the program.

Reference: "The JASON Project: Creation and Evaluation" by Richard Kool and David Denning, Minds in Motion, Fall 1989, pp. 45-57.

Although hundreds of multimedia CD-ROM offerings are being developed and it is one of the most efficient storage mediums available, this market will probably be slow for the next several years. Presently there is just not a large base of customers, and the cost is high. Confusion is caused by competing systems (CD-ROM, CD-ROM XA, CD-I). Other stumbling blocks are that good software is only now becoming available, there has been a lack of standards for retrieval software, and there is a limited number of creative people who can develop these products. There has been a gap between the people who understand the technical end and those that understand the artistic end, making it an effort to produce multimedia products.

Reference: "Who's First? Chicken, Egg? CD-ROM Players or Programs" Computer Pictures. by Emily Leinfuss, Dec90/Jan91, pp. 53 - 56.

Animation

"Authoring software is to multimedia products what directors are to movies. It lets you combine and orchestrate the different media elements you're using into a coherent, sequenced whole." (p. 54) HyperCard is the most widely used, and MacroMind Director, the best high-end product, but there are also many others. Some authoring packages are also animation packages, and vice versa.

At the bottom line, animation is a succession of still images that the mind interprets as motion because of what is called persistence of vision. Slower than 16-18 fps (frames per second) the phenomena breaks down and we perceive flicker. TV films usually run at 24 fps, professional video at 30 fps, and a standard Mac monitor repaints the display at 66 fps.

Reference: "Authoring, Modeling, and Animation" by Lon McQuillin, MacUser, Feb. 1991, pp. 52-68 in Buyer's Guide Supplement.

Traditionally there have been two approaches to computer-based animation: hypermedia software and the frame-by-frame animation software. However as time goes on, the line separating these approaches is blurring. (p. 558)

The word "animation" means "bring to life", however most people associate it with movement. Due to an effect called the persistence of vision, where minimally at 15 fps (frames per second) and optimally from 24 to 30 fps, animation can be achieved by any time-varying alteration, such as light sources, motion dynamics, shape, color, transparency, texture, etc. Traditional character-based animation was developed as an industry by the Walt Disney studio between 1925 and 1930 as a two-dimensional drawing technique. The screening of the first Mickey Mouse cartoon was an industry landmark. Director Ray Harryhausen is credited most often with clay-based 3-D stop action animation. (p. 559)

Many of the new techniques in desktop-based animation incorporated 3-D modeling and rendering software. Objects are drawn and described in a 3-D modeling program such as AutoCAD, Super 3D, Swivel 3D, or MacroMind 3-D, as wireframes or smooth-surface models, and assigned ranges of motion. A standard file format is written - PICS, (Mac), .DFX (PC) and that file is imported into a rendering program where the drawings are translated into realistic objects using mapping, shading, ray tracing, and autotracing. Rendering is still extremely slow on a personal computer. (p. 560)

"Computer-assisted animation has caused a revolution in the entertainment industry - nearly all of the steps in a sequence can be done on computers. In fact, animators were one of the early adopters of computer technology, minicomputers, and workstations." (p. 561) Even though personal computers don't have the power or storage space to render scenes realistically or interpolate complex tweening for professional animation smoothly, they are making progress. Presently they are used as prototyping stations in the film industry for the more advanced animation work done on powerful computers. (p. 561)

Reference: Beyond the Desktop: Tools and Technology for Computer Publishing, by Barrie Sosinsky, New York: Bantam Books, pp. 549 - 597.

There are problems animating on the Mac. A full-color, 24 bit, 640X480 pixel image represents nearly 1 meg of data. No Mac is designed to display 20 of those images in a second. However in more limited ways, the Mac succeeds very well. For example by slowing down the entire process and functioning as a slide show, or by animating only small elements in the foreground (called sprites) and leaving the background the same.

Special effects, such as wipes and dissolves, help "to display text information and animate charts to convey information more clearly." (p. 55) Animation is also clearly ideal for showing processes that one image alone can not do, for example, how an engine functions.

The two most common output formats for Mac-based multimedia displays are the Mac itself and VHS-cassette videotape. Direct animated display on the Mac is constrained by limits of image quality and movement but offers the potential for interactive control.

Output to tape can provide sophisticated image complexity and quality, however there is a loss of interactivity.

- HyperCard - probably the most used program; version 2.0 lets you create interactive animated presentations of surprising complexity. Its main problem for animation is that it replaces entire screens which slows it down and takes a lot of memory. But you can use XCMDs to run animations developed elsewhere.
- ADDmotion - installs seamlessly into HyperCard 2.0 and adds the ability to do sprite-based animation. Animations saved in the stack run without having to include any program with the stack. Studio/1 - a low-priced package that works in black and white only, but its tools are innovative and easy to use.
- SuperCard - which is able to play PICs files created by 3-D drawing programs, and move graphic elements around the screen.
- Plus 2.0 - prime advantage is that it works across platforms (stacks created on the Mac can be opened on the PC).
- Authorware - the only completely object-oriented multimedia authoring tool for the Mac; includes MacRecorder in the package.
- MediaTracks - helpful in creating Mac-related training.
- MacroMind MediaMaker- people who are not multimedia specialists can string together a series of picture icons (picons).
- MacroMind Director - creates, stores, and plays back complex animations, disadvantage is the 8-bit color.
- FilmMaker - easier than Director, but wide range of abilities.
- Animation Stand- sophisticated program with own language
- interFace -actors are animated talking heads.
- MouseRecorder- records & narrates live on-screen presentations.

Animation cont.

Points to remember if a presentation is to be videotaped:

1. Avoid horizontal lines that are an odd number of pixels wide.
2. Avoid abrupt transitions between primary colors (especially reds) as saturated colors tend to smear on an NTSC monitor.
3. Use software that performs anti-aliasing for graphics and text. While anti-aliasing creates much better looking NTSC images, it often results in perceptive blurring when the same image is viewed directly on the Mac. (p. 60)

Simple animation and authoring can be done on anything from a humble Mac Plus on up. But without a more sophisticated platform, you come up against a number of problems, including color, speed, screen real estate, and output options. An ideal system for creating animation and doing authoring would be a Mac IIx with a 600 meg hard disk drive, an 8-mm tape or optical disc drive for backup, a multisync monitor, a 24-bit color display card, a frame-grabber card, and possibly various video sources - VCRs, laserdisc players, and cameras. All this easily adds up to \$20,000 or more.

Reference: "Authoring, Modeling, and Animation" by Lon McQuillin, MacUser, Feb. 1991, pp. 52-68 in Buyer's Guide Supplement.

Working with Images

In a course offered by the Ontario College of Art students are introduced to computer graphics tools including the excellent icon and object oriented front end of HyperCard. They created an animation-like series of interactive graphics that combined together to create two stacks, one a "high-end" graphics for an interactive animated game, and the other designed for presentation.

Reference: "HyperCard in an Artistic Environment" by Isabel de Correa Hoffmann, Minds in Motion, Fall 1988, pp. 27-31.

Graphics in HyperCard need to be in a MacPaint format (72dpi, B/W) and can be imported using the Import Picture command available when in the paint mode, or brought in using the clipboard or the scrapbook. (pp. 325-328)

In order to display color or gray scale images, HyperCard 2.0 has a special XCMD called Picture. Images can be PICT, MacPaint, or PICT resources. Limitations are that the images are in external windows and also the image itself can not be manipulated within HyperCard. Use of these takes clever scripting and prior preparation, and results can create stacks in which viewers will not be aware of color limitations. (p. 365)

Reference: MultiMedia Design with HyperCard by Stephen Wilson, Englewood Cliffs, NJ: Prentice Hall, 1991.

Picture research is conducted with a particular subject or image in mind, or with a message or feeling in mind. Either way relies on catalogs and collective memories of staff people. The sheer number of items desired, the variety and complexity of queries cries out for some type of automated access. Unlike books, pictures do not bear their own identification. There is the further problem that words ultimately cannot describe or represent an image but the image can't be represented without them. Present attempts at cataloging images have been most disappointing and proven difficult for researchers to use, while being costly and labor intensive.

Videodisks offer rapid access to images and can be linked to cataloging data.

Reference: "The Library of Congress Non-Print Optical Disk Pilot Program" by Elisabeth Betz Parker, Information Technology and Libraries, Dec 85, p. 289-299.

The struggle to catalog images now has a three decade history (the 60's, 70's and the 80's) and as we move into the 90's this has become "like the legendary pursuit of the Holy Grail." (p. 117) The striving has resulted in increased sophistication, but still no code. The first phase of computer-assisted art object cataloging was characterized by large-scale multi-institutional systems. The 70's saw groups developing cataloging for specific purposes, and multiplicity developed. The 80's saw a harmonization of systems take place, identifying a core of useful data elements. The goal is no longer as clear as it was 20 years ago, but the quest continues.

Reference: "The Quest for a Code" by Deidre C. Stam, Beyond the Book, edited by Toni Petersen and Pat Molholt, (Boston, G.K. & Hall, 1990), p. 117-143.

Missing Page

Chapter 3: Journal

May 1991

After attending two Graphic Design Archive meetings, Professor Roger Remington and I discussed possibilities for a thesis project for myself. I indicated that I would be interested in designing an interface that could be used as the basis for creating a presentation. Such a project could be connected to the GDA. Subsequently I also discussed the ideas with Professors Mark Collien and Jim Ver Hague.

The Initial Idea

Sept. 9, 1991

I met with Professor Collien to discuss my ideas as to the direction of the project. I explained to him that I envisioned creating a program which would enable the user to get the images off the disc into the computer, creating a presentation that could be either computer based and independent of the videodisc or converted into slides and thus completely independent of the computer itself.

**Reshaping
the Idea**

He saw two disadvantages to the idea; first, the picture files would get too large to be easily manipulated or transported (on diskettes) and secondly, the quality of the images would be diminished. Instead he suggested leaving the images on the video disc, but having the titles and other information on the computer screen.

There would still need to be a tool to find the images of various designers and save them as a set, a way to print text as handouts, the capability to edit presentations, a method of digitizing of thumbnails (for computers with the right type of board) and a way of arranging by titles.

He suggested looking at user profiles and getting feedback as to what kind of tool would be most helpful. The research, evaluation, and graphic standards should all be taken into account before beginning.

Sept. 10, 1991

Professor Ver Hague looked at a list of goals I had made for this project. There seemed to be one primary goal, and all the other points supported that one. Once we had established that the purpose of the goal is to answer the question "What?" then the rest of the list fell into place as objectives. For they seemed to answer the question "How?"

**Establishing
Goals and
Objectives**

Sept. 12, 1991

When I met with Professor Remington, I showed him my proposed first draft of the goals of the project. He studied it and made a number of helpful recommendations.

Today, at the GDA Meeting, the first draft of my thesis proposal was presented to the committee and suggestions invited. Susan Williams, from the Media Resource Center, suggested I stop over and take a look at a tool they presently have in the Resource Center. It is a Fox-based database that she felt was impressive in its use. She said at present there were primarily two systems being developed to store and access images, one used more by librarians, and the other more by art museums.

Issues of Availability

Later, I talked with Professor Barbara Hodik. She expressed enthusiasm for the project, while saying her first concern was how it would (could) be available to her as an instructor. The most convenient placement would of course be within her own office; however she did not have Macintosh equipment, and neither do the majority of teachers, not to mention the extra equipment necessary to use the videodisc. While it did not seem likely that the school would immediately set up every office in such a way, the second best solution seemed to be to have a Mac computer with the necessary hardware dedicated for staff use at some point within the College of Fine Arts. Once again, there is no way of knowing at this time if that can be implemented. The third best solution is to have a station set up in the IMD area, perhaps in (near) the archival area. This would allow people who were interested in the archives to have access to the programs developed within the entire GDA project.

The second main point that Barbara spoke to was the eventuality of having a library where images were stored digitally; in other words, a digital "slide" library. She suggested consulting with the slide librarian as we may wish to design our program so that it could be intentionally compatible with slide-image digitization.

Need for Tutorial/ Introduction

The third main point she brought up was that she felt there had to be some type of introduction that would allow people to get used to using the Mac. She felt that since a Mac was not a computer that she used, or that indeed, many used with any regularity, directions would be needed for using the mouse, how to move it, how to select, etc. When you are used to typing commands into the computer, the concept of moving a mouse and clicking seems totally foreign and at first very awkward. She suggested to embed into the introduction the techniques, commands, and activities that would be the very ones needed to use the actual module(s).

Also there needs to be some kind of introduction to the archive so that a person new to the concept can get an overview of what is available and an understanding of the project itself. Perhaps a simulated training module could take a person through two or three of the most probable uses, like creating a lecture on a specific designer, and thus provide an example.

As to the program itself, she felt that it would be most helpful if it were menu driven; in other words, that a designer would be picked from a list of possibilities, and the next options chosen from another menu, and then further options, as a tree branching out.

Incorporation of Sound

As she read through the sheet containing the primary goal and the list of objectives, she particularly liked the idea of creating a multimedia presentation, and hoped that meant incorporating sound. She also suggested that there might be music to choose from for the opening sequence or perhaps the designers themselves speaking on some aspect of their work. She referred to a tape of an interview from Herbert Bayer, in particular. It would be nice to be able to excerpt segments that would fit into the presentation and have them attached to a particular part of the presentation.

There are always tradeoffs, she felt, and that the quality of an image may have to be sacrificed in order to show it to a large group. Creating a presentation that had to be viewed by huddling a small group of people around a little screen was of absolutely no use for her. She referred to the Barco projector and making use of its ability to convert video tapes to full screen, but she definitely wanted a full screen capability. Another thing she wanted to see was both the image and the text together at the same time, so that the text could emphasize the points of the image being shown.

As to the handouts, she would like to be able to have it print out as a bare outline for the students to fill in, and also have the capability to enter the entire presentation for her use as a presenter.

September 30, 1991

Professor Collien explained to me that since the inception of the GDA it has been an idea to create a presentation module. Some of the concepts that need to be considered for this module are that it is designed to create a presentation that can stand alone; that it can be used by student and/or instructor; that it gives instructor/student tools to make a presentation easy to assemble; that it contains a mini-authoring system; and that it can integrate text and images into presentation.

**Design
Considerations**

We discussed use of software for this project, ruling out MacroMind as a primary program since its authoring platform is rather complex. SuperCard was considered for its color capabilities and stand alone creations; however, because of the color, it runs slower. In the end it was felt HyperCard would probably be the best program for our purposes because the rest of the GDA project is in HyperCard; color can be added into HyperCard; and it is an authoring tool with which I am familiar.

Finally we talked about the external command (XCMD) "ShowPict" in order to introduce color into HyperCard Presentations and converting PICT files to PICT resources.

October 1, 1991

At the Media Resource Center I met with Susan Williams. There I had the opportunity to observe the GDA system as it was set up to be available to students. We discussed the idea of categorizing images and the dream of electronic database for images. We also spoke about the Barco projector and she explained to me that there is a training video available to help instructors make use of it.

**Observing
Present
Facilities**

October 3, 1991

Professor Hodik attended the GDA meeting today, and explained that in teaching her classes of 45 to 100 students at a time in a large room, she had developed what she termed "survivor thinking." For her the real need was being able to put something together in a reasonably fast and efficient way. Her time was very limited in terms of finding materials and images.

Faculty Input

She stated that she would like to see things done where students could put together "TV" papers in which they could pull together images into a presentation. There should also be a field where the teacher could make comments.

Continuing the Research

On my own I began examining HyperCard stacks that show presentations that have been made or presentation-maker types, including "Legacy of Graphic Design." Also I have been and am continuing to do readings of references on multimedia and related areas, and I created a HyperCard stack to keep the notes in orderly fashion.

In order to learn the use of color in HyperCard using the PICT Resources and "ShowPict" command, I spent some time experimenting and also examining the scripts in the ColorPict stack. I also worked on a flowchart for Stack Builder and Stack Templates for Graphic Design Style project. This project (although intended primarily for student use) will serve as a starting point for my thesis project.

In preparing for a meeting with Professor Hodik to demonstrate a number of possible features that could be considered for a presentation maker format, I constructed two presentations. One was made in HyperCard using interactivity, color, and text capabilities, and the other was made in Aldus Persuasion using its outline format and templates.

Initial Demonstration

October 10, 1991

Professor Hodik was amazed by the capabilities of the Mac in the handling of image files. I demonstrated for her an interactive HyperCard project, and a project done in Persuasion. She asked how the image files were developed in the first place. So I showed her the scanner, what a file looks like after it is scanned, and then the use of PhotoShop to adjust the colors and shading to match the original image. She told me that she was tremendously inspired.

November 11, 1991

I reported at the GDA meeting that the dates and assignments for the thesis shows have been made. The third show, of which I am to be a part, is set for April 20-May 7, 1992 with the reception scheduled at 7pm on the 24th of April.

December 3, 1991

Today Professor Hodik she told me that she had applied for a productivity grant to study how to make a course more interactive (distance learning with the computer, she called it). She also had applied for a leave of absense to give her a chance to visit other campuses and observe steps being taken in that direction. Both were granted.

She spoke of the difficulty of trying to translate electronic projects into a classroom setting. Computer projects by their very nature are usually intended for the individual in order to be interactive, and as soon as you try to extend the project for a group, the interactive element is lost. However she would like to redesign mass audience courses to be more interactive.

Prototype: StackBuilder

December 10, 1991

Professor Remington and I first discussed the StackBuilder project for the Graphic Styles. I had a problem using the long list of options from which the user chose the style. It did not work with the external function (XFCN) "Pop-up Menu" and neither did it work with a scrolling locked field. For the time being, the user must type in the name into a field rather than pick it out of a list.

In terms of icon design, I can script open the icon editor, but cannot script to affect what happens after the editor is opened. That means, that in terms of programming to help the student in choices and options, I really cannot affect anything that happens while the editor is open.

Professor Remington pointed out that the use of the help button on the main panel (available on all screens) seemed to be redundant in that I had help options available along the way.

Part way into our meeting, Talos Tsui, a senior graphic design major, came to meet with us, as he will be using the project in a first attempt to make it more workable. He was interested in knowing how to put colored pictures into HyperCard stacks; we set up a meeting for Wednesday, 12-18, at 12:30pm to further introduce him to the project as far as it is, to get his ideas and input, and to give him the commands for showing picture resource files.

At noon I met briefly with Professor Collien and explained to him the difficulty that I was having in scripting to work within the icon editor. He said that unless you knew programming and could program an XFCN to do the task, it was not possible.

Later Professor Ver Hague and I went over the draft of the time line that I had made (A-1). We also discussed the components of the thesis, so I could be sure that I had taken into account the entire project in designing the time line. I showed him the research that I had completed to date (which is organized in a HyperCard stack). My next major focus will need to be to work on the flowchart of the project.

I stopped in to see Professor Hodik to let her know of the progress on the project. I mentioned my frustration of the fall quarter, seemingly talking all quarter and not accomplishing anything noteworthy. She said, it was all part of the process, and even though I thought I had not accomplished anything, the very fact that various aspects were considered and discussed was a progress in itself. Thinking about it, I realized that as I was beginning the actual project design, I did have a much clearer idea of where I wanted to be going, so the process was working more than I had thought.

I showed her a summary of the project in terms of the "need-to" and "wish-to" parts. Some of the grand ideas for the project design may be like the programming within the icon editor, just impossible to do in terms of my present knowledge and the capabilities of the program. She didn't seem at all upset and said that that was as valid a part of the thesis (which Jim, too, has told me many times) as was being successful. The thesis can document what has been tried, what works and what does not (which she called a "null hypothesis.")

She asked what I needed from her to continue. I already knew that she wanted me to concentrate on designers, Bill Golden and Memmed Femhy Agha. She gave me a brief outline of how she approaches presentations on designers, and it was agreed that I would do some research, come up with a text file (containing a title, main points, and notes) and relate it to a particular image. When I get the information together, I can get back to her to check for changes and verification.

Initial Steps

The Process is the Progress

All these people are very busy, and I appreciate their willingness to take time and meet with me. I came away from these meetings with new ideas and enthusiasm.

December 12, 1991

Considerations for the Flow Chart

Once again I continue to be impressed with the importance of the process of feedback. Today in the GDA meeting, I presented my first draft of the flowchart (Appendix B-1) featuring the main parts of my project. Once this overview flowchart seems to be holding together then each of the three main parts will need to be broken down into their component parts.

Professor Collien asked about whether I had decided not to use images off the videodisc since it wasn't included here. Actually this was a complete oversight on my part, and should be included. Professor Ver Hague thought that people might rather enter text right into the stack rather than using a separate word processor to create a text file. Professor Collien added that if the user were to make one error somewhere in the file, the entire import process of the text could be "messed up."

Personally, I like the idea of creating text in a word processor where there is a spell checker and to me it seems rather a logical thing to put in the separators (delimiters) and then import it, but not everyone thinks the way I do. At the time of the meeting, I conceded that the user should be able to enter text just from the stack if that was desired, and that there could be a choice. Now after a little time has gone by, and considering how confused things can get if the user makes errors in the creation of the text file, I think perhaps I will not add the import text feature, but start with the assumption that the user will enter the text right in HyperCard. I can always add the option to import text at a later time or even just to my own copy.

Templates: User-Designed or Self-Designed?

We also talked about the template for the stack. Initially I had thought to have several templates available to the user's choice, but then recalled my use of the templates in Aldus Persuasion. Even though there are about 45 templates, I can never seem to find any that fit my mood or the project, so I usually end up creating my own. It was felt that the user should have the option to design his/her own template as the program began. This is one place where I think we are really going to feel the lack of color with HyperCard's limitations to black and white pixels. So instead of offering the user a wide range of choices for templates, we will give the user a tool, such as Template Editor or Template Designer, in which the user can control field placement, font choices, buttons, etc.

Revisions for Flow Chart

So these suggestions will lead to some revisions on my flowchart, which all things considered, I think are necessary and offer needed improvements. Professor Remington asked me what I planned to have ready for our next meeting, which will be after Christmas break. As I looked at my Time Schedule, I realized that not only did the revisions affect my flowchart, but that they need to be incorporated into my time schedule as well. For the week of January 6, I had hoped to be working on the importing of the text, but now that has eliminated. Since I noticed that I had not included the design of the templates, let alone creating a Template Designer, I decided that perhaps creating the tools for designing the templates was the place to start.

January 16, 1992

Designing Menus

During the Christmas break I worked at trying to create functional menus in HyperCard. My first two attempts were not successful. I found a menuMaker in HyperCard toolbox, and could create menus using that, but I wanted to take it further and be able to understand the script so I could script a menu when and wherever it was needed.

After studying the script of a menu that was generated by menuMaker, I created my first successful menu. It was short and called "Food," having a list of three foods. When one was selected, its name came up in a field.

My next step was to take a list of graphic design styles that Professor Remington gave me for the StackBuilder prototype and see if I could make a menu that would take over 30 items and still work. Again, I had it feed a name into a field. This time there were no problems and it was successful. I assume if I can feed a name into a field, I will also be able to put in other scripts as needed. This to me was an important step in working on my thesis as my actual project will need to be relatively transparent (invisible and not in the way for the user) while the user is creating a presentation. Menus may be the way to make it work.

The following week (Jan. 6-10) I turned my attention to a project that I was doing for Professor Bob Keough (specifically the organizing of a large quantity of clip art in HyperCard). I was attempting to create a number of tools that allowed the user to easily move between stacks of organized clip art, copy, move, and create their own stack of art for a specific project. I created the buttons and worked on the scripts first, and later began working with some of the actual clip art. Because it was set up to have the buttons in the background (to avoid duplicating them for every card), the artwork completely covered them up.

Need for Palettes

To move these functions on top of the artwork, but still have them somewhat movable, I decided to try to create a palette that contained the needed functions. I pulled up the "Navigator" palette, because many of the functions contained there would also be very helpful for this project, and made a list of the functions that I had already developed specifically for manipulating the clip art. I used Capture to duplicate the familiar icons on my palette, and decided to use type (the standard Geneva bold used for HyperCard buttons) for the specialized functions that I developed. I ended up creating a number of palettes before I got the look to be just right. Under each button was a simpler handler that referenced a script in the stack script. My next task was to develop the handlers.

The simpler scripts worked right away, while some of the more complicated functions still have some bugs that need to be worked out. There seems to be a definite implication from this project for my thesis. When the entire screen is being used to create the presentation, the placement tool box (the part I am to create for the user) must be carefully thought out. One answer to consider is that it could reside in a menu that gave a list of choices, calling up appropriate palettes and dialog boxes to collect necessary information or complete specific tasks.

Amalgamation of my Learnings

I shared with Professor Ver Hague that it seemed like so many of the projects on which I had been working had ramifications for each other. I also expressed to Professor Remington my thoughts that so many of my classes and the work that I was doing seemed to be coming together. He suggested that I not let those thoughts get away, so I will complete them.

My last AVI course was Communication Theory in which my project was to investigate the difficult concept of categorizing visual images. Professor John Ciampa, my instructor, provided me with some resources, as did Susan Williams from the Media Resource Center. My project was in the form of a HyperCard stack which collected information (people's opinions about ten specific images) and collated it in the back of the stack. The project made clear to me how very difficult it is to organize visual images for easy retrieval.

In the winter quarter, I began a project for Professor Bob Keough, mentioned above, in which clip art is being organized in some method to located an image without searching through each piece. Again the need for categorization, classification and retrieval of images has become important.

Then in my Electronic Photography course we are dealing with image acquisition or specifically, how images are moved into the computer and manipulated, including the use of Still Video Cameras, scanning 2D and 3D objects, and the scanning of transparencies and slides. Once again, this deals with imagery on the computer, which is more of a major component of multimedia than I had previously considered.

In my animation projects with Professor Ver Hague, I worked with adding sound to my projects. At that point I became aware of the vast world of the knowledge of sounds and sound files. Sound, being an important component of multimedia, may be somewhat overlooked. One needs to know how to capture a good quality clip, how to edit it, what legally can be used, etc. But beyond that, once the sound file exists, how can it be synchronized with the computer files? Different computers play at different speeds, so how does one keep the sound still in sync? What about sound boards; are speakers needed?

Implications for Multimedia

It seems to me that sometimes that multimedia is talked about, and the showiness of the computer gets in the way of what it really is all about. The front end of it is so dazzling with transitions and special effects that can be easily added with the computer, that the basic components of multimedia get overlooked. It is true here as in anything that a chain can never be better than its weakest link. Multimedia will never be better than the sum of its parts, so all of its pieces need to be considered in designing a multimedia presentation.

Revisions in Thinking

January 18, 1992

I extended the flowchart (Appendix B-4) as I thought that the introduction might be most effective in showing the multimedia aspects (incorporating images, graphic elements, text and sound) if it was done in MacroMind Director. Also considering that this project has to be in a thesis show, it would add a showy element to help it present well.

On a mouseclick in the introduction, users could be given the choice of starting the program or getting help. I think that the help also might be most effective if it was in MacroMind. On choosing the program, the template designer stack in HyperCard would be opened. There would be a number of templates already designed as choices, any of which could be modified, or the users could create their own.

After choosing or creating the artistic background, and making choices as to number of text fields and formatting of the text, the program could then create a stack for the user. Before leaving the designer section, the user could be asked for the resource files that would be used and these could be added to that stack, then that stack would be opened. At that point, the user would be in HyperCard with a one card stack, and the editing features would be choices on the menubar (to avoid getting in their way on the screen.) When the editing features were chosen, an appropriate palette with the choices could open. The menubar could also feature the output choices.

In addition to modifying the flowchart, I worked with picture resource windows to see how they would work in bringing color into a HyperCard stack. When the entire screen was filled with the picture window from a pict resource file, it covered and conflicted with the text fields. The cursor could find the field, and it could be typed in, but when done typing, it was covered up with the picture. However, a very small colored graphic that was not in the same proximity as a text field worked fine.

Disadvantages of the PICT resource files were that they could not be easily revised on the screen, and also their placement and use within HyperCard has to be scripted. This may make it more difficult to make the template design easily interactive for the user in terms of modifying a template design. Also they are more difficult to display than a PICT file as the PICT resource file must first be installed into the resource fork of the stack before it can be accessed from a script. It would not be easy to script that process so that it could be easily interactive for the user.

January 20, 1992

Professor Remington and I went over some of the ideas and changes in my flowchart. He had files for me on Agha and Golden, so I could begin working on a presentation for Barbara.

Later I showed Professor Ver Hague the revised flowchart and some of the other projects that I was working on that tied into my thesis. In the StackBuilder project I created a Menu, which provide the names of graphic styles. Then in the ImageViewer project I again used an added menu, this time to easily move between the stacks that make up the project.

In addition to the menus, I began working with a palette which I created using the PowerTools stack. The palette provided a great learning experience and took a bit of trial and error to get it working right, but has proven to be helpful for this particular project. I thought that perhaps the main part of my project might be a menu that brought up appropriate palettes, so that the actual screen wouldn't be all cluttered up, as it was the presentation. He mentioned an article that described how to create a palette and incorporate color into it. (He brought it for me the next day.)

Using Picture Resource Files

Menus and Palettes

GDA Meeting

January 22, 1992

During the course of the GDA meeting I presented my flowchart (Appendix B-4) and we discussed options. Professor Collien mentioned that there was an XCMD in HyperCard that lets you make little PICT files while you were in HyperCard (CLIPTOPICT). Professor Ver Hague brought up the thought that it might be easier to incorporate the images from the videodisc while still in the template designer, which I thought was a good idea. So I added that change to my flowchart (Appendix B-5).

Ideas for the Introduction

January 25, 1992

I have been trying to put my thoughts together as to the MacroMind introduction. Movement could be horizontal and vertical using graphics to precede visual images. Visual images could reflect the expanded knowledge that multimedia can pull together, and also the Graphic Design Archive, since this is also connected to that project. Ideas for imagery could include the vastness of space, the infinitely small world of the microscope, children working at a computer work station, perhaps an animation of a sphere turning that blends into an image of the earth as seen from a spatial view, something in nature, and imagery from the graphic art world.

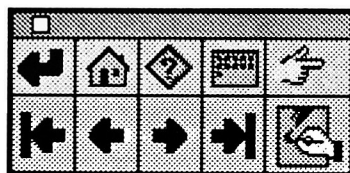
Progress Report

February 24, 1992

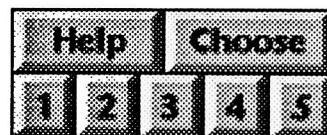
I submitted to Professor Ver Hague an evaluation of my current progress on my thesis project, including what had been accomplished and what I had hoped to accomplish by this date (Appendix A-3). While I have not made the progress that I had hoped, the time schedule was ambitious. Next quarter I should be able to focus on this project more completely.

Palette Redesign

March 8, 1992



Trial palette based on the Navigator palette.



First "Chooser" palette,

My first task was to redesign the palette that came up with the opening window. The original (Appendix C-1) was based on the Navigator palette in HyperCard and had a combination of arrows and icons. Since the number of templates was going to be about 5, I designed one that contained a row of buttons across the bottom for the numbers of the templates (instead of using the arrows.) Above the numbered buttons were two buttons, the first with the word "Help" and the second with the word "Choose."

After installing this into the stack, and putting the scripts in the Stack script window, I began noticing all kinds of problems. HyperCard didn't function right at all. I would go to pick a tool and I would be taken to another card. It didn't take long to realize that the words that I had picked for handlers for the "Help" and "Choose" buttons ('help' & 'choose') were causing problems as they had special meaning within HyperCard. I went back to the paletteMaker and changed the handler names within the buttons and reinstalled the palette. The problems seemed to be gone.

March 10, 1992

As I was showing Professor Remington the project I could barely manage to make it work. Once again I was having all kinds of problems with HyperCard. I was suspicious then about the reinstalled palette. I wondered if instead of replacing it, HyperCard had installed another alongside the original containing the errors in the handler names.

Palette Problems

I asked him about a rationale for the design of the templates. Considering the design elements as well as the functionality of the templates, how could they best be planned? He suggested perhaps using a continuum from a template designed for all text to one designed for a large image with steps along the way. I thought that to be an excellent idea.

Rationale for Template Design

Another suggestion he gave me was to develop a grid that could be turned on or off to establish some organization to all of the layouts. He gave me two references: one on the use of grids in layouts and one on the use of type. Keeping the number of characters in a line to between 40-50 will improve readability. He also suggested that I look for information on screen design and layout for screens. More specifically he felt that the very narrow graphic elements on the screen could be just a bit wider.

When I met with Professor Ver Hague, I told him of my idea to add black and white bitmaps under the colored graphic elements, so that when the cards were printed the graphic would show. He once again brought up the problem that with a screen design white lettering on a black background is more effective than black type on white. We talked about ways of doing this and how some of them were not reliable. I said the best way I knew was to completely cover the screen with a transparent button with the hilite set to true, but then this did not allow anything on the screen to be accessed. We started working on creating a button that could turn on and off at the click of the mouse, and after a short amount of scripting, got it to work. We were both pleased with the result.

Inverting the Screen

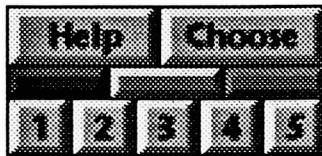
Clicking on the screen reversed the entire screen, so that the text was now white on a black background. At this point fields could not be typed into, but the palettes were still on top and functional. Clicking on the screen again hid the button, and the type changed to black type on a white screen and at this point could be edited.

At home, I used resCopy and found that there were actually three palettes installed in my stack, where there should at this point only be one. I removed the two older versions and the project functioned as it should.

Fixing Palette Problems

One of the problems in using PICT resources for graphic elements is that when the card is printed in HyperCard, the PICT resources do not appear. I decided to solve this problem by creating a black and white bitmap for each screen graphic that could be pasted beneath the picture Resource image on the card. In addition I made another colored version of it, so that for each graphic there is a bitmap, a red version, and a blue version. Having the graphic in two colors and the bitmap gives the user three choices for each graphic element.

Creation of the Graphic Elements



Second "Chooser" palette.

Back in HyperCard, I realized that I had to provide the user some way to toggle between the bitmap and the colors. In PaletteMaker I redesigned my palette (Appendix C-1, #3), so that a middle row of colored buttons was added. Blue was at the left, gray in the middle and red at the right. Then in the stack script I

created the code that would switch the colors for just the first layout. It took some working to get that to work, but at this point the blue button brought up the blue picture resource file, the red button changed it to the red one, and gray interrupted the showPict command (in an 'on idle' handler) and allowed the bitmap to show.

Disk Error

On backing up for the evening, I was informed that the file could not be copied due to a disk error. In the process, my backup was erased. I assumed that the little floppy had been written to and rewritten so many times that it needed some cleanup. I opened MacTools and began the Optimizer, and indeed files were fragmented. I ran the optimizer program defragmenting the files and cleaning up the disk, then tried copying my thesis file again. Again I got a disk error.

At this point I realized that the error was in the original, and now I didn't even have a recent backup, which was destroyed first by the attempted save, and then completely wiped out with the optimization. I once again opened the Optimizer and this time ran it on the cartridge containing the thesis project. I was informed that optimization could not be completed as there was a problem on the cartridge, to use the Rescue program to fix the disk.

Attempted Rescue

I ran Rescue, and was informed that it would take 50 minutes to check the disk for bad blocks, and clicked OK. After about 40 minutes I was informed that a bad block had been found in "Thesis4," which was the current project file, and was asked if I wanted the file copied to another location so that the block could be marked. Again I clicked OK. The file "Copy of Thesis4" was copied to the hard drive, and checking for bad blocks continued. At the end of the 50 minutes, only the one bad block had been found and marked. I tried running the Optimizer again, but this time informed that there was another error and to run Rescue another time. This took another 50 minutes and this time turned up nothing. I went back to Optimizer and this time was informed that there was an error in the catalog of the disk. I ran Disk First Aid (from Macintosh) which found the problem and fixed it. This time when I ran Optimizer, it ran; the cartridge's fragmented files were put back together, programs moved to the beginning of the cartridge and it was cleaned up.

I then copied "Copy of Thesis4" back to the cartridge and tried opening it. HyperCard loaded and my first screen came up. At this point there is always some delay as the stack script loads the palette and picture resource files for the first screen. As the picture resource files displayed, all of a sudden everything stopped, the top picture looked like someone had smeared across the bottom of it, and the computer was completely locked. It was very late, so I turned off the machine and went to bed.

March 11, 1992

It occurred to me in the morning that because the error was in just one block, and the stack seemed to load fine until the display of the showPict file, that perhaps I could locate the error using ResEdit. In ResEdit, I opened "Copy of Thesis4" and also an older version "Thesis3." First I located the code for the showPict XCMD and compared the two files, and could find nothing dissimilar. Then I decided to look at the PICT files and opened the PICT files in "Copy..." On opening the picture used for the opening screen, the picture was distorted (like a smear across the bottom), the computer locked and everything had to be restarted. I started ResEdit again and this time I first got the identical, but intact picture out of "Thesis3" and without opening the window for that picture in "Copy of...", I pasted it in. When I opened my project this time, it opened and worked fine.

**Program
Recovery
with ResEdit**

I decided to take the scripts that were working for the first card in toggling between the blue, the red and the black and white version, and copy them onto the other 4 layout cards. To my surprise two of the cards worked fine, and the colors toggled, but on the other two cards the toggle didn't seem to have any effect. I worked with the message tracer, the debugger, and variable watcher to try to trace the problem. After several hours I could still not discover it, so I made copies of the scripts so I could consult with someone on the problem at school the next day.

**Color Toggle
Problem**

Next I worked on the layout and grid. I created a grid of five rectangles across and five rectangles down for the screen. Through the work I had done with the toggling of the picture Resources, I had already discovered that pictures could easily be turned on and off. I put the grid on the background and made a small button that I put in the bottom right corner that allowed the showPict of the background to be set to the opposite (not the showPict), so the grid easily toggled on and off.

**Creation
of the Grid**

My next task was to begin to create some functionality within the portion of the program that would allow the user to choose font, style, size, etc. I used a card with a rectangle of that card to indicate the screen, and the choices for the fonts for the title and body text below. As a choice was made that button highlighted, and the corresponding text reflected the change. Also the choice was recorded in a field that for the time being I put on the last card, the Help card.

**First Attempt
at Layout
Design**

March 12, 1992

On waking up I had a hunch that I knew what was causing my problems on the two screens that did not operate correctly in toggling between the two colors and the bitmap. There had to be a script at a deeper level that was interfering. I opened up the program, and went to each of the screens and looked at the background script. On both of the cards (each has a unique background) there was an 'on idle' handler. Removing those scripts eliminated the problem completely and the toggle worked.

**Color Toggle
Problem Solved**

I wanted to give the user some control in screen design and tried two approaches. The first one made use of two thumbnails of layouts on the left, and clicking on a thumbnail would cause the elements in the diagram on the right to move.

**Second
Attempt
at Layout
Design**

In the second approach all the possibilities of placement for each element were considered. On choosing an element (the title, for example) all the possible locations for that element would appear on a thumbnail. Clicking on any of those locations would cause that element to move to that spot. After making both of these options workable, I called my husband (who knows a little about this project) to look at the screen and tell me what he thought it was for. He had no clue, so I realized that the meaning and functionality was not being communicated, and the design and layout would have to be reassessed. Unfortunately, there wasn't time for another approach before the GDA meeting today.

GDA Meeting

I printed out the stack as it was for the meeting today, and left early because of the blizzard conditions. At the GDA meeting Professor Remington talked about Edward Tufte's information graphics and his "flatland" terminology. He presented a map proposal for the NGDA in introducing Cindy's project. Cindy shared ideas for her project and approaches that she was trying.

In sharing my progress I handed out the stack copies showing my latest work. There were two very real problems here. It is extremely difficult to show the interactivity on flat paper, and I ended up making lengthy explanations to help people begin to have an idea of how things were interrelated. The second huge problem was due to the fact that neither the PICT nor PICT resource files print in HyperCard (which was why I added the black and white bitmaps for the PICT resource files), so the color in the image is totally lost. Due to these factors it was hard to share the functionality of what I was doing. In retrospect, I should have made screen dumps, converted them to TIFF files and printed out of another program.

Feedback

While I was interested in concentrating on functionality and design, most of the comments were on the actual text (which I only added incidentally to show that text would be there.) It is hard to add the exact text until the program is up and working properly, then the text can be made to fit the situation. Specific comments were that the word "template" was confusing and not understandable to a person outside the design field as was the word "grid." The word "layout" was suggested for "template." Instead of "Text for Title," use "Title Text" and instead of "Text Placeholder..." use "Body Text." The use of templates was questioned altogether, thinking that all the presentations need everything and full accessibility of all the options should be given. At the same time things were too complicated and needed to be simplified. A grid should not be used at all because people outside of the design field won't even know what it is. Professor Ver Hague remarked that if people didn't know how to use a grid, all the more reason that one should be used and the elements conform to it, as they would have a more professional look to their presentation with a minimum of effort and understanding on their part.

I expressed my frustration of dealing with the unknown images that would be brought in, not knowing their size or if they would be horizontal, vertical or square in shape, and how best to deal with that. Professor Remington suggested that examples be set up based on the largest and/or most complex situation, because then they could handle any of the other alternatives.

In response to the lack of understanding of what was happening in the stack design, I wanted to rethink the strategies and approaches. I made a list of ideas to consider. The first idea was to start out by getting the name of the presentation, then allow the user to choose the number and type of fields, and graphic elements, and place them as desired. This could be done in a stack that has been created for the user. That would mean that all changes and choices are saved within that stack and would not have to be moved or copied into a stack at a later time. The primary stack could store all the instructions, resources, etc. that need to be used. Control could be given to the user through palettes and menus. The second idea I had was to keep the five layouts and use selected thumbnails to provide limited choices to the user, and an easier and quicker access, so that much of the layout work and design is already done. Finally the last idea I considered about was to provide access to a palette which will allow the user to choose the graphic element and place it to their specification.

March 17, 1992

I showed Professor Remington Thesis4 and the revisions I had made in Thesis5. Unfortunately when the palette came up, there was no PICT file in it. It was completely blank. However the buttons still worked (since I knew where they were). He liked the use of the grid and showed me how to align things when working with type in relation to it. (Align text left - ragged right, align to top of grid, ragged bottoms.) He questioned giving a choice of four fonts and suggested maybe going down to two, a serif and a sans serif. For the meeting on Thursday he suggested I re-evaluate my progress and bring in something to show the current workings of the stack.

**The Blank
Palette**

Later that morning Professor Ver Hague and I discussed several problems. The 36 point size of the types is not always a nice bitmap, and they will have to be generated or loaded into the program. Presently I was offering a choice of four point sizes, and many are just not working well. We decided to offer 24 and 36 point for the title, and give no choice for the body copy, setting it at 18 point. Also eliminate the choice of alignment for the text in the fields, using flush left for everything. He suggested that I see if the leading or line spacing could be set in relationship to the grid to create a nicer look.

**Narrowing
Font Choices**

Also since currently the pictures are brought in as picture resources, he suggested that I change over to PICT files as that is what will have to be done for the actual presentation.

I made a list of the next tasks to be accomplished. The arrows that move the Title and Text fields need to be made functional. A font program needs to be used to generate 36 point bitmap type in plain and bold. The title, text and director palettes as well as the graphic element on layout 5 should be redone. It would be a good idea to check about leading or line spacing and relationship to grid. A way should be created to get palettes back if they are closed by mistake. Professor Remington suggested creating a method of tracing the user's path and saving it. Also I would like to finish writing chapter 1 of the thesis and update my progress and time chart.

Layout of the Palettes

March 19, 1992

By today, I had written scripts for the up and down arrows of title field. These were complicated enough that I changed the scripts so that both the title field and the text field could use the same handler, so the scripts wouldn't have to be duplicated.

Helvetica	Plain	↑
Avant Garde	Bold	→
Times	Italic	←
Bookman	24 36	↓

Title Field Palette.

I revised the title field palette (Appendix C-3,#1) to include four font choices, plain, bold and italic, 24 and 36 point sizes, and the directional arrows. I left italic as a choice because in the bigger point sizes, it didn't look too bad and also I needed another line to make the palette come out nicely.

Helvetica	Plain	
Avant Garde	Bold	
Times	↑	→
Bookman	←	↓

Textbody Field Palette.

The textbody field palette (Appendix C-3,#3) was revised to include four font choices, plain and bold, and the four directional arrows.

TitleText Field	Help
BodyText Field	Save
Image Location	Load
New Layout	Quit
View	View

Bitmap for Director Palette.

The Director palette (Appendix C-2,#1) was revised to include a button to indicate completion (not scripted yet), the title and body text field palette buttons, choose a new layout (not scripted yet), the toggles for black & white screens and the grid, and the help, save, and quit buttons. The "Load" button was eliminated because if they had a file they wanted to load, one would assume that the user would open it instead of a template designer.

I decided that pictures should be not be dealt with in this part of the program, as each image could be so different from others that it would not be possible to make a general layout that would work for most images. Rather, the tools will need to be provided on the screen at the time the image is brought into the presentation.

Currently the new palettes are all in black and white bitmap, having been created in PowerTools by cut and pasting together the original palettes. The graphic pictures for these need to be recreated in PhotoShop and saved as PICT resources. Then they can be entered into the PICT files in the resource fork of the stack using ResEdit.

Revisions and Experiments

I created an on mouseUp handler in the stack script that checks if the palettes have closed by mistake and for the current location of the user, and then shows the appropriate palette for that spot. If either palette is open, it resets the location of that palette to the mouse location because now that I changed from picture resource windows to PICT windows, the PICT windows can cover up the palettes.

While I tried working with leading or line spacing and relationship to grid, I didn't have much success in being able to program it or even change it yet. I created a field that keeps a record of the user's path and then writes it to a file called "History" and the current user number on quitting. On revising the time chart (Appendix A-4), it has become apparent to me that my initial goals were overly optimistic. The scope of this project covers so much programming and the involvement of so much control that to complete the various facets each in a week was extremely unlikely. It now appears that I may be only able to get the first part of this done, if I am to do it well. I also revised again the writing of chapter one: the plan and proposal.

March 19, 1992

At the GDA meeting I explained the organization of my project to date, as well as having a copy of what had been accomplished, and what was yet left to do. Professor Remington suggested that I trace every move the user makes in the history tracer section. As to why the palette came in empty, I said that I had read that if 32 bit QuickDraw was not running and the PICT files had been saved in 32 bit format, that could create an empty palette. Professor Ver Hague said that for the type of graphics that I was working with, 8 bit should work fine.

GDA Meeting

I asked Professor Collien about the massive script that is currently being compiled in my stack script, and how to create a new stack without the entire stack script, as many of the scripts will not be needed. He suggested having the scripts for the new stack in a field and those could be scripted into the stack script of the new stack.

Another problem that I brought up was the fact that the PICT window was covering the palette. Professor Collien felt that the layer of the window could be controlled. He suggested looking into the documentation of HyperCard 2 (that was put out when HyperCard 2 first came out in the fall of 1990) as that had the best documentation on windows.

March 20, 1992

After reading and rereading the documentation on HyperCard 2.0 related to windows and palettes, I did some experimenting with setting the rectangle of the windows and trying to control their layer. I didn't have any success with the layer of the window, and in fact, couldn't find documentation on it in any of the references I have available to me. I think that the rectangle of the window can be set, which means that on importing the PICT files, the pictures can be constrained to match the grid size. How well this will work, however, remains to be seen.

**Scripting
for the
Palettes**

I also tried programming the "New Layout" button which is on the Director palette. It is supposed to check to see if either the TextBody palette or the Title palette is open, and close them if open. Then it closes the Director palette, shows the Chooser palette and changes from a layout card to a template card. However, something in the "if..then" statements checking the TextBody and Title palettes causes the script to stop. (In programming the palettes there are no error messages if there is a mistake in the script; instead nothing happens. That is the only clue that the script is not working. Sometimes it works not at all, and other times it goes so far and then quits.)

**Control of the
PICT Windows**

Scripting the Movement of the Fields

March 21, 1992

I spent my time working on finishing the programming of the movement arrows on the two field palettes. I had first programmed it for one field, and when the program became quite lengthy, decided to make use of a global "moveMe" into which the object being moved could be entered. The directional arrow moves the field according to the grid by getting the height or width of the object depending on whether the movement is horizontal or vertical. Once the scripts were worked out, it functions quite nicely.

Show/Hide Palettes

March 22, 1992

I worked again on the New Layout button today, trying to use hide and show commands. If on opening the stack, the palette is opened, but hidden, then the New Layout button could cause the palettes to change from being hidden to showing. Not taking into account that the palette may be closed, this didn't work well. I thought about using a global for a switch, but ran into a problem on how to capture the "close palette" message if the "close box" is clicked on the palette, so the global switch could be changed.

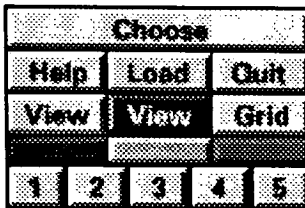
Redesigning the Palettes

I spent most of my time today working on the four palettes that are now used in Template7. They have been patched together graphically up to this time, and I have worked on the programming, but not on their specific design and layout (done in PhotoShop). I have already discovered in using the palette that putting the "Quit" button close to other buttons that are used often can cause the "Quit" button to be easily hit by accident. So I moved it to an out of the way location. Also I wanted to make the buttons uniform in size within one palette and between the various palettes as well as making the palettes conform to the grid size that was chosen for the templates.

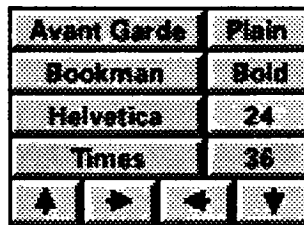
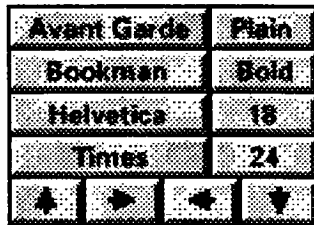
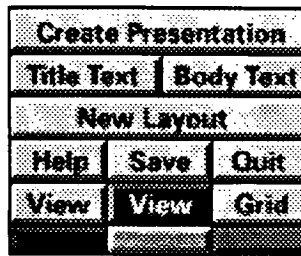
I began in the PowerTools stack of HyperCard. I have made my own copy of it, so that all the various palettes can be stored, and it is easy for me to move the work I have done on them with me. A palette is made up of two layers: the bitmap graphic beneath a layer of buttons. The graphic is the only thing that shows in the palette, and the buttons are the only thing that functions. The two elements need to be put together carefully to create the palette.

I started with the black and white bitmap in Power Tools . First I changed the height and width of the buttons to fit within a section of the grid. I then checked to see if the words would still fit on the buttons in a Bold style font, and found that they would, but the font size needed to be reduced from 12 point to 10 point. I tried to create uniform sizes of buttons and rows, and have them organized in a logical manner.

Then I moved the buttons on top of the graphics. Initially, I worked with transparent buttons so I could see the graphic beneath the buttons. This caused a problem as it was difficult to see the positions of the buttons. So I changed to laying opaque buttons over the top of the graphics. It was easy to line up the buttons and see exactly where they lay. The disadvantage now was that the graphic layer of the palette could not be seen. So there was no perfect answer, but it was easier to get good results if the buttons were opaque.



PICT files created for the palettes
in PhotoShop.



I closed out of HyperCard and opened ResEdit. Using ResEdit I opened the HyperCard stack in which the palettes had been installed (Template7), and opened the PICT file. In the PICT window all the graphics files are included, and it is a simple matter to scroll through them and locate the bitmaps that have been saved for the palettes. I opened each of their files and copied them and pasted them to the scrapbook, then quit ResEdit.

Now I opened PhotoShop and one by one, copied the bitmapped palette files into windows. I changed each window to RGB color and applied colors and anti-aliased text to create buttons with a three-dimensional effect. Then each file was saved as a Picture Resource file in 8 bit using the system palette. (Saving in 32 bit here will cause the palette to appear as empty if the computer does not have 32 bit QuickDraw installed in the system files.)

Back in ResEdit, it was just a simple matter of opening up each of the picture resource files and the HyperCard stack containing the palettes. Then each of the PICT images from the picture resource files were pasted into the location containing the bitmap image in the HyperCard stack. Now when that stack is opened and the palettes are called up, the picture within that palette should be the one that was created in color and not the bitmap.

March 23, 1992

Again I tried working on the New Layout button, building off the global created by HyperCard for each palette when HyperCard calls it up. The problem with that is that there is no global in existence until the first time the palette is opened, and if a palette has not been opened then the script ceases running at that point. However, even if they were both open, it still didn't seem to work. In addition I now noticed that the palette scripts that changed the text styles from bold to plain was no longer working, and I couldn't detect any reason for them not to be fine. Obviously something was interfering.

Clarifying Issues

March 24, 1992

I expressed my concern to Professor Remington about the possibility that I may not be able to finish what we had at first set out as my goals within the time allotted. Roger was very supportive and shared with me the idea that it was perhaps this type of time crunch that made one focus on what was truly important. He suggested that I focus down to the bare minimum of basic functionality, and work on a time plan in relation to a revised flow chart.

As we looked at Template7 (all the windows and graphics now came in fine) other suggestions were that the vertical graphic in Layout 5 could be changed to a thin horizontal line positioned at the top of the grid above the title. The words "text body" suggest paragraphs and maybe need to be changed; perhaps to "Heading" and "Subheading."

Later I mentioned my frustration to Professor Ver Hague that I seemed to be stuck on the programming for the template designer part of the thesis. I was not close to the actual presentation maker, and there were still parts of the template designer section that were incomplete or not working. Jim suggested that I choose one of the layouts to work with and move into the presentationMaker section, and then posed a question on the use of templates. Should the templates be able to be manipulated by the user or should the user simply have a number of choices and not be able to manipulate them at all? It occurred to me that herein lay a major problem for where we were at this point in time in this thesis. Why not have the template section just allow the user to choose a template? All the manipulation could be done by the user in the actual presentationMaker section of the program. He agreed that this would be a good approach.

Documentation of Progress

March 24, 1992

Before making such major changes on this project, I worked on updating the flowcharts. First, I worked to correctly translate the ideas and concepts of Template7 into an updated flowchart. Then I worked on a flowchart where the Input part of the program be simply a matter of few choices (picking a layout and graphic color) and then after generating a stack for the user, all the remaining design decisions and tools are in that stack.

This meant some major changes to the program. The chooser palette needs only Choose, Help, Quit, the three color buttons and the five number buttons for the layouts. Basically the stack is the opening card, which gives an overview and brief explanation of the palette, and displays the palette, the five layout cards, and a help card, which again gives more in depth information as to the chooser palette and the workings of the program. The three color buttons will allow a user to choose either red, blue or a black and white version of the graphic.

The power of the chooser palette will be in the Choose button. This button will first check that the user is on a layout card. If so, the layout number and the graphic color choices will be recorded, then a new stack script (which is stored in a field which the user does not see) created. After getting a name for the presentation, it will create the new stack, insert the new script into the new stack, and make sure the correct palettes and resources are moved into it. "Choose" also must copy the layout card into the new stack, open the stack, and finally, make sure the Director palette is open and chooser palette closed.

Also the use of the grid in the background as a graphic with the bitmapped version of the colored graphic in the front layer will need to be changed. The bitmap version needs to be moved to the back layer because the user may wish to cut and paste bitmapped graphics in the front layer of the card. However, at present if the bitmap graphic were moved to the back layer, the script which toggles the background grid on and off would also toggle the bitmap graphic on and off. So the grid needs to be accomplished in some other way than turning on and off the background graphic. My initial idea was to use buttons that showed and then would be hidden to reveal the graphic.

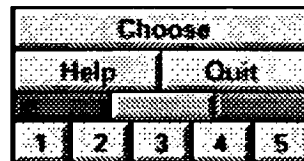
Because so much functionality was built into Template7, my next job was to separate the functions. The functions of each of the palettes had to be clearly defined and the scripts moved to the appropriate places (stack script or card script waiting to be copied into the new stack.) I spent some time in the redesign and plan of the functions and layouts, and from that made a list of tasks in order to accomplish the changes anticipated. Using this list as the basis, I then redesigned my time schedule (Appendix A-5).

March 25, 1992

I began the development of TheTemplate8 in which I deleted all the previous palettes. I also deleted unnecessary cards and scripts in the stack script. Trying to create a new method of getting the grid to toggle on and off, I used the show and hide commands for buttons, but then found that buttons that are changed from opaque to transparent work wonderfully in hiding or revealing the grid beneath them.

March 27, 1992

I worked on a new graphic in a horizontal position for layout 5; also redesigned the graphic for the chooser palette. On the last card of the stack I created a field that would hold the new script for the new stack.



In scripting the Choose button, I had trouble again because when an error occurs, a palette script just stops, without returning any error messages. I tried two things to help myself out. First I put in messages so that I could tell just where the script was when it stopped by what the last message in the message box was. (Eg. Put "I have just created the stack.") Even more helpful was to move the script to a button on the card, because there it did return the error messages, making it easier to trace down a problem. Then when the script seemed to be working, move it back to the stack scripts (where the palette scripts are located.)

Starting TheTemplate8

Scripting with Palettes

There was no need to worry about copying over the resource files in the resource fork into the newly created stack. When the script creates a new stack, it copies all the resources of the parent stack into the new stack. Since all those resources are not needed, I experimented for a while working on a script to remove them. In Power Tools I used the scripts from the "Remove Resources" button as my guide, and then tried to script the new stack so that some of the resources were removed. I worked on variations of the script for a while but wasn't able to get it to work.

Inserting a stack script

It took some doing before I successfully got the new script copied into the new stack. It still is not 100 percent reliable. Some of the problems were caused by the width of the field where the script is stored being less than the width of the window where the scripts were typed, which caused returns to be entered in the wrong places. That made the scripts unworkable. Other problems had to do with scripts either on cards or in the stack that were interfering with the scripts from the choose button.

The function of the choose button is first check to make sure that the user is on a layout card, and not the help or title card. If that requirement is met, then the layout number and graphic color choices are recorded. A new stack script has to be created for it from the field on the last card of the stack (which the user does not see), and the particulars (the graphic color and layout number) are fed into it. The entire script is then written into a variable, which takes about three minutes because of the length of the script. The user is asked for a name for the presentation, and the new stack is created. The new script is inserted and the layout and help cards are copied into the new stack, and then the stack opened.

By the end of the evening, the process had worked twice, so I knew I was getting close.

March 28, 1992

Palette Revisions

I worked on the palettes for Image, and Viewer. First I worked in Power Tools to create the palettes and install them in the stack. Then I worked in Photoshop to create the colored graphics, and installed them into the palettes using ResEdit, as described previously.



I continued working on the Choose button's script, and found that some of the scripts in the stack and on the cards were interfering. I removed the history scripts, and changed the card script to a one line handler. The scripts for that handler could be made to be appropriate for which ever stack the card was located in. I seem to be making progress, but sometimes it is so painfully slow.

March 29, 1992

Scripting Modifications

There have been a number of small problems that I worked at tracing. The palette for the images didn't come up, which was easily solved as the script didn't have the correct name. In the process of creating the new stack, the computer asks for the path to Template8. On investigating I found I had used "long name" instead of "short name" to refer to Template8.

Because it took so long to copy the script for the new stack, I added icons of a little clock whose hand spins around, so that it looks like something is happening during those two minutes on the screen. I rewrote the scripts for opening the palettes, so that if they are clicked on a second time (and the palette is open) it will close. The buttons that display the palettes now work like toggle switches.

In creating the new stack, the chooser palette gets pasted on top of itself, so I added a script on closeStack, to close the chooser palette. I added to the script for the choose button to have the help card copied into the new presentation stack. On the help card in the new stack, I added a field that updates itself when the card is opened, creating a list of the cards currently in the stack.

As I tried out the add card button, I noticed that there was no provision made for navigation, so I went back to Power Tools and redesigned the palette with arrows, which allow the user to move forward and backward. I also added to the add card scripts so the script asks for a card name, and names the card for the user. There is still a problem with the graphics, they are not showing consistently, and not always at the right time. Moving and changing the scripts have made a muddle of the original organization and workings of them.

March 31, 1992

Professor Remington and I discussed the changes in the palettes in terms of their functionality. For some reason when I opened the stack, the chooser palette would not appear by itself, and we got an error message "no such window", so I brought the palette up by using the message box. Then when I demonstrated the creation of a new stack, the process only partially completed and the script stopped. So I opened a presentation that had been created previously, and again, had trouble with the Director palette. Once the palettes were opened, they worked fine. The text sometimes appeared in an unusual font, so the need to get the bitmaps of the fonts loaded into the stack as resources was reinforced.

We also discussed how to create the interface between the laserdisc (where the GDA images are stored) and the presentation itself. The Mac image needs to be connected with the laserdisc image number, and all "under the hood" as Professor Remington said. I asked about connecting it to the query card that has already been created. The problem with that, he told me, is that the query card only accesses some of the images, and using that would limit what images the user would have available to him/her. He suggested perhaps looking at the query used in prototype 2.0 or 3.1, or even considering using a wide open index. This would provide access to the entire disk. He showed me a disk controller card and its functionality.

I explained to Professor Ver Hague the problem I had been having with the windows and the graphics. On looking at the scripts, we noticed that the location of the image was put into the global pictLoc, so that pictLoc contained both the x and y coordinate (eg. "12,118".) When the location of the window was set at pictLoc, nothing happened. But if pictLoc was picked apart: "put item 1 of pictLoc into xLoc"; "put item 2 of pictLoc into yLoc"; and then the location set at xLoc,yLoc, then the picture showed properly.

Demonstration Problems

Interfacing with the Archive

Picture Window Problems

We talked about the interface with the GDA laserdisc, and Professor Ver Hague suggested that I get the scripting from projects that had been done already. I mentioned that perhaps a field that could be created to read off the presentation stack and create a list of the screens created for the presentation. Then the user could be given a way of matching the Mac screens with the videodisc number of the image desired.

Image Acquisition

I also worked on acquiring image that have to do with Agha and Golden for the demonstration presentation that I hope to complete for the thesis. Once the images were scanned, I took them into photoShop where I resized them to correspond to the size of the grid in their width dimension and let the height dimension fall where it may. I also adjusted the color and brightness, and cropped them as needed.

Controlling Picture Windows

April 1, 1992

My goal today was to try to get the scripts from TheTemplate8 completely working. It has been tricky to control all the windows, as some are PICT resource files and some are PICT files. The picture resource is the colored graphic that changes in this stack from card to card, and the PICT file image also must change from card to card. The help card must have no graphic and layout 3 has no graphic, but must support a PICT file window. Then in the created presentation stack, the picture resource (colored graphic) must stay the same on all the presentation cards with the PICT file changing from card to card, and no image on the help card. So the scripts must be able to determine the appropriate image. I used the global "cdName" to check the name of a card to determine if a picture resource file was to appear. In scripting, if the short name of this card is the same as cdName, then show the picture resource file appropriate to that card. On opening each card, if it is to have a picture resource file, the card name is entered into cdName, while on the screens in this stack where there is to be none, cdName is not updated, and the image does not appear. That seems to be working now in a reliable fashion.

PICT Window Problems

I began with the Picture command to bring in some of the Golden and Agha images to the layouts. The images saved in gray scale (8 bit) changed the palette and graphic colors to gray. I had to resave each image using an 8 bit system palette, and then this problem was eliminated. Another problem that developed was that the closing of a card sometimes generated an error message: "no such window." I looked at the variable watcher, and it contained the name of the window, so I still haven't exactly tracked down why this error occurred. As I worked on tightening up the programming for the windows, I found that I got that error less. It may be due to the fact that in testing the choose button script repeatedly the script was interrupted and not completed, so that the variables governing the display of the picture window were not completely reset.

I also started to get in some of the information on the help card for this stack. At this point I am not especially pleased with the way it looks, particularly the bitmap graphic, but it's a start. Additionally I began gathering all of the files that will make up chapter 3 of my thesis report: the journal. I moved the files sequentially by date into QuarkXPress, and set up a format. Some of the information fits into the dated journal approach, but I think that most of the charts and timeline schedules should be moved into the appendix.

April 2, 1992

Replacing Stack and Card Scripts

I spent the first part of the morning tackling the script for the Choose button. It has turned out to be somewhat more complex than merely making a new stack and copying into it the layout card. When the new stack is made, it has the stack script of the "parent" stack. Because the presentation stack requires other palettes and other programming, that script must be entered. Before it can be entered, it must be updated with the correct layout number and the chosen graphic color, and then it is all set to run the presentation stack.

That process produces a new stack with the background of the selected layout, but the card is completely empty, containing none of the necessary fields or buttons. So the next step is to go back to the "parent" stack and copy the chosen layout card and paste it in the new stack. Initially, when that was completed, I would get error statements as the card would contain scripts on it that made no sense in its new location and were unrecognized in the new stack. I revised the card script so that it called a handler in the stack script, and the handler then could be appropriately designed to fit each stack and thus the problem would be eliminated. That worked fine and the error statement was eliminated.

Problems arose when I tried to insert a new script into a card when it was in another stack. After replacing the entire stack script, it would have seemed that the same programming would work, but I could not seem to get a script into a card in a different stack. So I took another approach, and copied the existing card script into a variable called temporary, and then inserted the new cardscript onto the card before copying the card into the new stack. Then back in the parent stack, the original card script saved in temporary was restored to the card. That worked perfectly, but when you consider that the entire stack script was replaced, it seems odd that I couldn't also replace a card script.

I started working on the script for the image palette that would import a picture. This meant an updated script had to be generated for each card as the image was imported, because in the presentation part, each picture needs to be uniquely associated with each card. I created a field which contained the card script for the cards, so when a PICT file was imported, the name of the PICT file was inserted into that script. Then the updated script was made to replace the existing card script. Once again I had problems with getting the windows to close on leaving a card, and had to insert a new handler (on closecard) into the stack script. It worked with the same globals, but this time closed the picture window. Some of the problems still remained. Again, I suspected that it had to do with the process of debugging a script. When an error statement was generated and the script discontinued, all the variables may not have been set correctly.

Importing a PICT file

Another possibility needs to be taken into account, and that is that the user may want to change the PICT file. The script must check to see if there is a PICT file on the card already. This should be fairly easy, as the global pictState either contains "true" (a picture window exists) or "false" (no window.) It will need to ask if the current picture should be replaced, and if the answer is affirmative, then close the existing picture before replacing it.

GDA Meeting

At today's GDA meeting I showed them the actual working of the various palettes, and how things connected together on the Mac in the computer lab. One issue brought up was in the use of dummy text, but specific images. Two approaches were discussed. One was to replace the dummy text with words related to the images, while in the other the images should be replaced with grayed out versions (kind of a dummy image.) Another idea was to gray out the image area and then dissolve into an actual sample, so that the user could get an idea of what an actual one would look like. With the complexity of the programming on the cards, I hated to add anything like that in HyperCard, but maybe that would be suitable for a type of introduction in MacroMind Director.

Susan Williams brought up a point that has concerned me before. The user cannot tell the size of the text field. She suggested greeking (filling the field completely with dummy text.) I hated to do that because the clean look of the presentation would be destroyed and the field would never be used completely filled.

Approaches to Moving Cards

I mentioned the problems I was having in creating the script for moving a card. If the script cuts the card then there must be a way for the user to indicate where the card goes. If the user moves to a new location in the stack, how can "Paste Here" be indicated? Then, will the user know that the card will be pasted after the current card? One idea was to have two palettes that changed the button from "Move" to "Paste." Another suggestion was to create a card which read and listed all the current cards, so that by using fields and buttons, the list could be rearranged, and thus the stack. That way the user can see and work on the overview of the entire stack.

"Cancel" in the Dialogue Box

April 3, 1992

I began the day by updating the journal entries and then made my "to do" list for the weekend. The first task I began was to add verification for the Choose button. On entering the name for the presentation, the cancel button does not cancel the process. Instead it returns empty, cancels the window, and entered the default name, allowing the script to continue.

Revision of Layout Screens

I moved to PhotoShop, and changed the image areas for the picture files to boxes made of a border of a black line, and white line and a black line, so that whether it was viewed on a white or a black background, a line would appear. Inside the box, I used a graduated gray, and then with a black line, drew an X from corner to corner. Back in Template 8, I changed the scripts to bring in the new windows. At the same time I changed the wording of the text in the fields to be reflective of that particular layout and try to accommodate the size of the field, providing information to the user. Also on each of the layout screens I added a field for the entry of text. I put it over the subtitle field, so that the picture wouldn't cover it up. Presently it is hidden, and a button needs to be made that will show it so the user can enter text within it. I put two hidden buttons that toggle the grid on and off and also that reverse the screen color, so that I could have those options easily within use. (They were formerly on the chooser palette, but now appear only on the director palette, whose scripts are not available here.)

A problem developed in quitting HyperCard. On closing I was getting an error, "no such window," which when traced was from the closeCard script: "if pictstate contains true, close window myWindow. Put false into pictstate." Evidently the "doMenu quit" automatically closed all the windows, so by adding the script in the quit handler: "put false into pictstate" the problem was eliminated.

I moved the script waiting to be pasted in to the cards of the new stack onto the help card for the new stack. That way only one updating process was needed, and then the revised script moved with the new stack. This script is used by the images palette as it updates and pastes the script into the card as images are added or moved.

I began working on the scripts for moving an image by trying to script an image window to move up within a button script. Once I got that to work, I then revised the script for moving the image down, and then moving right and left. After I could make the image move, I then added the programming that would rewrite the card script, so that on openCard, the right picture at the chosen location would appear. This worked fine on one card, but then in moving to the next card, it didn't work. It seemed that all that was needed was to add quote marks around the name of the window, and then it worked fine.

In pasting the stack script from the presentation stack back into theTemplate stack, I received the message that the clipboard is too large to paste. I cut out a great deal of the comments in the script in an attempt made it shorter. Now that the image window can move around, it sometimes completely covered the field for the notes, making it impossible to type into it.

April 4, 1992

I began working on the HelpGuide card (the help card that is pasted into the new stack to function as its help card.) I wanted it to function as both a help card and since it already contained the list of the cards as the card used to move or change the order of the screens. I made 2 fields covering the cardScript, containing the directions for either help or move functions. For the help part, I used mouseWithin scripts on buttons that overlaid a bitmap version of all of the palettes. On a mouseWithin a certain button, all fields, but the appropriate one, would be closed.

**First Attempt
on the
Help Card**

Then I added a row of buttons containing numbers beside the field containing the list of all the cards. When move was chosen the help features were hidden, and a move and a rename button would appear. If move was chosen, further little arrow buttons would show to indicate where the card could be moved to. I worked at this for some time, and never got it working well.

April 6, 1992

At the Media Resource Center, Susan Williams showed me the videodisc of the National Gallery of Art containing 1645 paintings, sculptures, drawings and prints that are indexed. A HyperCard stack facilitated the search and provided for marking cards, adding the screen to a slide show . viewing marked cards, retracing your path, searching from a menu, or just browsing. It was interesting to see their interface.

**Observing the
"National
Gallery of Art"**

I looked through Ed Walker's stacks to find where he had done his scripting for the videodisc and copied that card, so that I could study it and use the scripting. I tried to work on ideas for how to best manage the process, and felt that it could perhaps be facilitated by the use of a field, as on the help card, that contained a list of all the screens, and another field that would contain the laserdisc number, if one had been chosen. We also need a field that will have the contents of the laserdisc, to make the browsing a bit easier.

April 7, 1992

Consultations

I showed Professor Remington the updated project. Once again, the palette would crash the scripts (no such window.) and then the picture windows would not appear. It was disappointing to have none of the windows show.

He again suggested something for the wall for the thesis show; I am hesitant on this as a flat representation of an interactive piece loses a lot, and is not truly representative. He also suggested a realignment of the indentation of the text on the subtitle fields of the layout cards. I showed him the help cards that I was working on, and indicated my extreme disappointment in their visual appearance. I asked him what he would think if I were to use the layout and graphics of the title card as the basis. He also suggested dividing the card into two cards) We also talked about the possibility of a visual introduction.

Later I showed Professor Ver Hague the newest changes. He too liked the idea of using the title card layout and graphics as the background for the help cards, and also thought that the help card would be more effective if the functionality of Help and Move were separated into two cards.

Another problem we discussed was the field for the notes, and the problem of the image appearing over it. As I told him about it, the answer seemed obvious: the button that brings up the text field has to determine if there is a window open, close it, and on closing the text field, then replace the window.

I told him about my idea for the opening introduction, that would serve as a visual tutorial as well as an opening. That should be a fun change from the heavy programming which has become the norm. So back home, once again, I have updated the journal, and now will make the new list of tasks that I want to accomplish for the next few days and see how far I get.

April 7, 1992

Laserdisc Data Files

I began by opening the shortened file that Professor Remington gave me of the catalog of items on the laser videodisc. First I alphabetized the items, then separated them into 2 separate lists: one of the topic and one of the start number on the laserdisc, so that the two lists were in a one to one relationship with each other.

Introduction in MacroMind Director

In MacroMind Director I began the animation for the opening sequence. My idea was to begin by creating the screen layout that is used as the title screen and the basis for the support cards (help, etc.) After the title screen is created by animating lines and shapes, then palette is brought onto the screen. I did this by creating a 3 dimensional rectangle in Swivel 3D, and

tweening it to make a PICS file. Then brought the PICS file into MacroMind and created a movie loop so that I could move the turning rectangle smoothly across the screen.

After the palette is in place, I made a replica of the browse tool, and simulated the choosing of the second template, then choosing the blue color and then clicking on the "choose" button. Following that was one example showing the opening screen from a presentation on Agha.

The music I used for the animation was recorded from Fresh Aire V, a tape by Mannheim Steamroller and the London Symphony. The excerpts were taken from a piece called Z-Row Gravity and were recorded and edited into soundLoops using SoundEdit.

April 8, 1992

I worked on the "Help/Move" card in Template8 and divided it onto 2 separate cards, so that the help function was contained on one and the move features on the other. This will help simplify the actual card, even though it means that two cards must be copied into the newly created stack instead of one. I used the graphics from the opening title card as the basis for the creation of its own unique background, and then used this background for the new help and move cards.

**Division of the
Help/Move Card**

Then the laserdisc card was moved onto that background as well as the help card belonging to template8. I then rescripted the color graphics that belong to the opening screen to appear whenever that background was used (on idle). I rewrote the text for the help card in template8 numerous times, each time trying to simplify and be more concise.

April 9, 1992

I reworked the timing of the opening animation, and redid the direction of the revealing of the text field on the opening screen. I also experimented with the placement of the sound files, whether to have them reside in the actual animation or in a separate sounds file. When they were a separate sounds file, they did not play in HyperCard, so I moved all the sounds back into the animation file. In HyperCard, I was having my own problems with the movie commands. The PlayMovie XCMD was no problem to install using resCopy, but the accompanying commands that can be added for more control never seemed to work.

**Rework of the
introduction**

At the GDA meeting Professor Remington told me that the module number of the PresentationMaker would be Prototype 3.2. After initially going over business, we moved into the computer room where I had the files loaded on the computer and ready to run. I showed the changes I had made in the help cards and the layout, and also the opening sequence, and explained the basic functionality of the project to date.

GDA Meeting

April 10, 1992

I created fields for the list of names from the laserdisc, the list of numbers for the laserdisc choices, the list of existing screens from the presentation, the numbers of laserdisc images to accompany the screens, and to indicate to the user the laserdisc image currently being viewed or chosen. I used a list of commands from the OMDR for the programming, and worked on adding the laserdisc images to the presentation easily.

**Laserdisc
Connection**

Continuing Palette Design

April 11, 1992

Today, I again redesigned the Director palette, adding a button to access the notes feature, and changing "ViewScreens," which was a reference to the final output of the presentation, to "Finish." I also worked on the palette that the Finish button would bring up, which I named it Presenter. To begin with, it has only 5 buttons: 1 forward, 1 backward, Quit, Edit, and Print. Edit closes the Presenter palette and restores the Director palette, so that the user can do more revisions. Print will provide the options of printing the actual screens or the text from the three fields that are on each designed screen (Title, Subtitles, and Notes.)

Title	Subtitle	
Image	Lazardisc	
Quit	Finish	Notes
Help	Invert	Grid
←	Add	Move →

Scripting Problems

April 13, 1992

It seemed today that no matter where I worked all I encountered was problems. I think the primary reasons for that are the reorganization of the numerous cards and fields, and that scripts are no longer necessarily scripted correctly. A change in one place had numerous ramifications, and it takes a while to track all of them down.

Some of the problems that I encountered are as follows. In HyperCard, no matter what scripting procedure I combined with PlayMovie, I could not get the movie to stop at the click of the mouse. I would always have to type Command-W on the keyboard to make it stop.

In working on Template8, the title graphics, as well as the windows, did not always show up reliably. The scripting of the picture windows seems to have deeper ramifications than one would initially think. In scripting the notes feature, the window has to be closed or it is over the field. No matter how I scripted the window, I could not easily get it back. Not only that, but all the other windows also disappeared, and there were no colored graphics anywhere. Like before I think the problem lies in the global pictState getting set to an off position and a script crashing in the middle, so that pictState is not being reset properly.

The invert button doesn't always work. It was initially set to work at the creation of the new stack when all the screens had a white background. Now that some of them have a black background, the button needs to test the background color to determine what will make the background changed.

On the move card screen, the last two cards need to be eliminated from the list of cards in the stack, as neither the help card or the move card should be renamed or moved. I added to the script, so that it would read all the cards minus two. The "Add" button needs to be reworked so that the user can cancel if desired. The cancel needs to be captured (if it is empty then exit handler.)

I tried the create stack button and it only worked to create the stack and add the layout card, then the script quit functioning.

April 14, 1992

Consultations

In meeting with Professor Remington, I began by going to the laserdisc card to see if the scripting would drive the laserdisc. It did not. He said that the OMDR commands may be different from those needed for the laser-disc. I said that I would go back to Ed Walker's stacks and see what I could find. There was also a major problem with fonts on this screen. The text appeared in a larger point size, causing the numbers in the videodisc fields to wrap. I showed him the redesigned Director palette and the newly created Presenter palette, and we also went over my statement for the thesis show.

When I met with Professor Ver Hague I showed him the opening animation, and told him the problems I was having with it. We solved it by putting a transparent button across the entire animation for its duration that was scripted "quit." It worked.

I showed him the changes in the palettes, the new look for the help and support cards, and the problems that I was having with the windows for the pictures. He too felt that the documentation on the use of picture windows was poor and it was difficult to find help. We also discussed the need for the child stack in determining whether it was in the editor mode or the presenter mode. A hidden field could store the information, and after being read as the stack opened, the appropriate palette could be displayed. Before we finished we listed a number of tasks to try to accomplish before the thesis show.

At home, I fixed the sound in the animation (where it had stopped abruptly, now it diminishes gradually), and corrected the timing of the choices made with the browse tool. In PhotoShop I fixed an error in the Presenter palette, where the pict file saved was a different size than the palette causing an empty white bar to appear across the top of the palette.

Fixing Problems

On the invert button, I corrected the programming, so that on each opencard, the information as to whether the covering button is hilited or not is fed into the global bgColor, and as a result, the button can function correctly.

I spent some time doing research on Agha and entered the information into the presentation. When I went to import pictures, all of a sudden nothing worked. As I followed the script with the variable and message watchers, even the local variables were not being read. So I quit HyperCard altogether, and restarted, and things went more smoothly again.

More Problems

The scripting for moving a card had to take into account for three situations. First, the user clicked before the first screen. In that case the card to be moved must be pasted at card one (which makes it card two), then card one must be cut and pasted, causing it to become card two.

The second possibility is that the user clicked in between the cards in the presentation. Moving a card in this situation is accomplished by going to the chosen location and backing up one card before pasting the cut card, as the paste will put the card after the current card.

The final situation that could occur is that the user clicked after the last card and maybe beyond the end of the current presentation. Since the global fldLines keeps track of the total number of lines, it was fairly easy to do a comparison of the choice and the global "fldLines," and use the least option for where to paste.

April 15, 1992

Generated Bitmap Fonts

Using Fontographer I opened the fonts for PresentationMaker and generated Helvetica, Avant Garde, Times, and also Helvetica Bold, Avant Garde Bold, and Times Bold, each in 18, 24, and 36 point sizes. There was no bitmap version of Bookman, so I looked up the laserfont possibilities and decided maybe New Century Schoolbook which did have a bold version would have to be used instead, so I generated both the bold and plain fonts in New Century Schoolbook. In Univers, I generated bitmaps for 10 and 12 point sizes.

I made a copy of PresentationMaker, before installing all of the bitmaps. The bitmaps added 280K to the size of the stack. I created a presentation, and opened both the title and subtitle palettes, and found that now nothing at all worked related to the font choices (changing between fonts, sizes, bold and plain). After spending nearly two hours in studying, generating and installing the fonts, that was most disheartening. In addition, when I used the Text commands to set the font in a field, displaying the generated bitmaps, they had a ragged appearance. So I made an identical field, and used the same font and point size, but in a font out of the system folder, and the difference was apparent. The generated fonts while definitely similar looked as if some little mouse had chewed on their edges. They didn't have the same sharp clean look of the original fonts. At this point, I felt that everything was better off before, the installation of the generated fonts, so I trashed the program and all the fonts, and then changed the copy I had made to the original.

April 16, 1992

GDA Meeting

At today's GDA meeting, I had my project loaded on the computer in the lab, so they saw the first pass at the introduction. It was pointed out that people from the country of Ukraine don't refer to it as "the Ukraine" which was on one of the presentation cards since Agha was born in (the) Ukraine. Professor Remington mentioned that he felt that somewhere toward the beginning of the movies, I ought to identify PresentationMaker with a statement to the effect "a tool for educators in Interactive Media Program Development." I noticed that while I had extended the movie, the quit button hadn't been copied all the way to the end of the movie. Also while I was showing them the actual Agha presentation that had been created, one of the picture files was missing, so the script stopped before its completion. The result was that all the picture files were turned off, and no picture windows at all would show.

April 17, 1992

Preparing for the Thesis Show

I spent a long morning at the gallery as walls were moved, and space allocated for the thesis show. I mounted the colored images of several screens onto the sign that I had designed, and returned to the gallery to hang it.

I reworked the "on idle" handler, discovering that I had extra globals that were not being used (MyColor, PictOn). MyColor stored the name of the selected color for the pict Resource, and PictOn kept track of whether a pict Resource file was on a card. Both of these were unneeded. The new script can manage well using only the global variables: pictName and pictLoc.

I then moved all the help cards onto the same background as the opening card for PresentationMaker, as somehow 2 backgrounds that were identical had been created. Now I will be able to create scripts to test for the background in order to determine whether the card is a help card or one that has been created in making a presentation. Also I eliminated the field where information relating to the layout choice was being stored, as it has become unnecessary.

April 18, 1992

Today I worked the Choose button, determined to eliminate any existing bugs and get that working correctly and reliably. I moved the "mode" field (which keeps track of whether the presentation stack is in the edit mode, so the Director palette should be available, or if it is in the presenter mode, which needs the Presenter palette) to the background of each of the layout cards and made it a hidden field. That way when the layout card is copied into the newly created stack and the stack script looks for the existing mode, it will always find it. Before when the field had been on the help card, the stack script searched for the mode on a card not yet pasted into the stack, causing the script to crash.

I found an error in the script where it looked for the new card script in the background field (no longer being used.) After eliminating the background field, and changing the script to read from the card field, a new stack was successfully created leaving the presentation stack intact with the card script properly replaced so that presentationMaker continued to function properly.

In MacroMind Director, I copied the quit button to the end of the animation, so that the user could quit anytime with just a click of the mouse. I also reset the sound scripting so that the volume faded out as the ending neared, and the fades corresponded with the changing of the images.

In PresentationMaker I worked on the help card for the new stack. There are six text fields, each immediately on top of each other, with six buttons below to hide and show the fields. I also programmed the top field to act like an index so that if the user clicked on a name, the appropriate field would show. I also wanted the appropriate palettes to appear when the help section referred to them (as a visual aid.) So I added the opening and closing of the various palettes as the appropriate help information was displayed.

Chooser cont.

I decided to get brave and move the Choose script out of its button (where it had been put for testing as there is no feedback when the handler is called from a palette) and to my great delight, it continued to work. As I ran Choose (the process of creating the new stack takes about 3 minutes) I made notes of changes to improve it: at the end of the script, change back to the browse tool; enter the name of the stack into the title field of the first card to be a visual cue for the user that the creation process is finished; delete the message line in the script that opens the message box with "stack has been created."

Controlling Palette Appearances

In the newly created stack when all the palettes are open in the edit mode, and then the user moves to the help card, the text is completely covered with the palettes. Because I already had a script that checked to see if each palette was open, and if it was, then close it, I moved those scripts into a separate handler, "paletteCloser," so I could also use that routine in opening this card.

That seemed to cause a problem with the Director palette now being on the screen twice, in two different places. When the palette is not already closed, if it is called for again from a script then another copy of the palette appears on the screen. Instead of calling for the palette on the openCard of this help card, I changed the script to check if the palette is open, and if so then set the location of it to the appropriate spot, and if it was not open, then to display the palette. That seemed to take care of that problem.

Progress Check

I took a moment to reassess my progress; at the present time the scripts in the PresentationMaker stack are all working efficiently, with the exception of the laserdisc screen, which has not been totally completed yet. The Laserdisc card needs to be scripted to copy the list from the presentation stack (in case one is in existence from previous work) and after completion of selecting the laserdisc images, the list needs to be copied back to the presentation stack. The Director palette has all the buttons with the exception of the Add and Move buttons working fine. Add and Move both need more work.

On the laserdisc card, I added controls to make the laserdisc function so it could be turned on and off, and created the buttons, fields and programming to choose a laserdisc player. As I worked on this, the thought occurred to me that in the presenter mode, there has to be an option to turn the disc player on, that some kind of connection must be established, so the program knows whether to read the laserdisc information and send the search command to the laserdisc for the image.

When I went to moveCard in the presentation stack, to move a card I got the message, "can't delete a protected card." On checking the card, I realized that because in the PresentationMaker stack all the cards are set to "cantDelete" for the protection of the stack, that property stays with the card as it is copied into the new stack. I looked up programming for cantDelete, and found that the script "set the cantDelete of this card to false" after the new card is pasted into the presentation stack, should take care of that problem without changing the setting on the original card.

April 20, 1992

Today I began with the image palette, and imported a PICT file. The image imported fine, but the card script was incorrect; the script lines are being fed into the wrong lines in the cardScript field. So I corrected the line numbers in the script, so that the correct lines receive the updated image information.

There's another problem here and I've noticed it before. When an image is imported to a screen, and the above mentioned lines are fed into the hidden card field on the help card, those lines show on the existing screen - even though it is a completely different card and a completely different background. Because in the current presentation the subtitles are on the right of the screen, this presentation text is mixed up with the scripting text. If I move to another card and come back, then the scripted text is gone and doesn't reappear. I went to the help card and moved the hidden field to the left. Now when an imported picture is brought onto the screen the picture window covers up the scripted text, but when the window is moved the scripted lines are there.

Phantom Script Problem

An idea that occurred to me was to set the name of the first card of the stack (the one that is pasted into the newly created stack to serve as its beginning) to the name given for the stack name. If the user decides another name would be more appropriate, it can easily be changed, but it is better than "Layout" and the number of the layout, which may not have any meaning in terms of the new presentation.

Naming the First Card

As I was working in a presentation stack in the presenter mode, and moved off of the help card, the Director palette came up instead of the presenter palette. I added to the script to read off the global that keeps track of what the current mode is for the stack to determine which is the correct palette to display. I also spent some time experimenting on the videodisc controller palette with the On and Off switch, and interconnecting that to the current player fields in terms of display. Though I have tried a number of approaches I am still not completely satisfied with the attempts.

Being now four days from the thesis show with tomorrow the last time to receive input from both my professors, I decided the wisest course of action would be to make a list of jobs that need to be completed, prioritize them and then attack them in that order. First I spent some time on the actual text in the help fields on the help card to make the information as clear and concise as I possibly could. On the laserdisc card, I added a set player option.

I experimented with different layouts for the presenter palette, and created the new layout in PhotoShop. I also redid all the palettes so that the coloring would be uniform and reorganized the Director palette. The four buttons that bring up palettes are all on the top and in a light blue color; the next two rows of buttons contain general functions and are a gray color, and the bottom row contains all the buttons that take you to a different place and are in a dark blue with white. This new arrangement has the advantage of making the palette slimmer. However this meant that I also had to open Power Tools and recreate that palette and reinstall it into both the presentationMaker stack and the Agha presentation.

Palette Revisions

Consultations

April 21, 1992

Today the conversations that I had with both Professor Remington and Professor Ver Hague centered around what exactly I needed to work on to be ready for the thesis show on Friday. In addition to having the presentationMaker working, it would also be nice to have the demonstration programs ready, so that the use of the various palettes could be demonstrated. Professor Remington suggested making notes of what happens at the opening, so that I could remember the comments.

With Professor Ver Hague, I brought up some specific programming problems that I was having. One had to do with the fact that when the program could not find a pict file that had been assigned to it, the script crashed and none of the pictures in the entire program would then show as the global was now set incorrectly. I mentioned some different ways that I thought we might be able to trap that mistake, but he thought it might be a HyperCard problem that could not be easily solved, and didn't want me spending a great deal of time working on it.

He indicated that he thought it was more important to get the fonts working, as the palettes still only work some of the time, for whatever reason. He mentioned that there are two ways in HyperCard to set a font; one is by selecting the Text option from the field window (or command T), and the other is by selecting the text and choosing the font from the menu. He felt that it was important to get the fonts installed into the stacks. Also since there was still some problem with the fonts, to use bitmaps wherever possible and eliminate any field/font problems.

We also talked about the problems I had encountered in trying to write the contents of the field storing the laserdisc numbers between stacks, and the pros and cons of just copying the laserdisc card into the new presentation's stack. In the end we decided that this might be the best solution, because then the new presentation could function completely independently of PresentationMaker.

April 22, 1992

Revisions to the Introduction

I again worked on the introduction in MacroMind Director adding the lines "a tool for Interactive Media Program Development" towards the beginning. Actually it worked out beautifully, much better than without it. The words come out of the white line to the right, and then as the white line pulls out across the right side of the screen and the words, they turn from white to black. The words dissolve off the screen, and the lines for the next part draw across the screen on the right half as before. All in all, the addition made a much stronger beginning both visually and informatively.

I had wanted to expand the animation, so that all of the layouts were viewed before one was chosen. Then after the layout was chosen, simulate the actual use of the palettes and their functions in creating the first screen. On bringing in the next layout I got a memory error, and the program crashed. I realized that the memory on my computer would not allow for all the pict files with the two sound files that already took up considerable space. I experimented by cutting the layouts into pieces and placing the pieces to try to save memory, but after doing that for the entire movie, I had only saved about 40K off of a file that was over 1000K. So I decided it would be best if I left it as it was.

I added to the Choose button script to copy the laserdisc card to the new stack. Then I had to track down the scripts in the new stack to change them from having two support cards (the help card and the move card) to now having three.

I also worked on the script for the laserdisc card for the slide show button, so that it would step through the selected images on the videodisc, highlight each line as the image was shown, and if there was no image on a line, then skip that line. I worked and experimented for a while trying a number of different approaches before I got something that succeeded. I had trouble making it skip an empty line.

April 23, 1992

Today I worked the fonts. Using ResCopy I opened the suitcase containing the Univers font. First there were a number of FONDs listed, each identified with its family name, such as Univers 55 and B Univers 65 Bold. Then there was a long list of NFNTs, making it impossible to tell which should be copied. (I really only wanted the 10 and 12 point sized for Univers.)

So I quit ResCopy and started the program ResEdit, and again opened the FOND "Univers 55." In that file the number for each of the point sizes and the variations were given. Each point size had four styles, number from 0 to 3. 0 represented the plain font, 1 was the bold bitmap, 2 contained the bitmap for the italic, and 3 the bitmap for the bold italic. This helps explain why bolded Univers and B Univers 65 Bold are not the same. In addition to the style number, there was also a Res ID number that identified that particular bitmap from the NFNT file. Once I understood the styles, I picked out the appropriate ID numbers. It was a simple matter to open the presentationMaker stack, and copy these resources into the stack. I followed the same procedure for all the other fonts, copying them in at 18 and 24 points. I am still in need of the bitmaps in the 36 point size. Helvetica and Times function fine scaling up to 36 points using ATM (Adobe Type Manager), but Avant Garde and Bookman use the 24 point font and spread it out producing a poor simulation of 36 point size type.

It was always a puzzlement why the two palettes that controlled the font choices, their sizes and styles should be so erratic in reliability of functioning. Before the thesis show I was determined to do whatever it took to achieve some accountability for those palettes. After considering the conversation I had with Professor Ver Hague about two ways to set the font for text, I rewrote the scripts for the font changes in the fields.

In addition to setting the field to that particular font or characteristic, I made a repeat loop that selected each word in the title and set that word to the font or characteristic. I tried it on a number of screens and every time it worked. Then I rewrote the script for the subtitle field, but because of the amount of words entered within that field there was a noticeable delay, so I changed that script to select the lines instead of the words. There was still some delay, but almost unnoticeable.

Working with Fonts

Another Approach for the Text Palettes

Laserdisc Slide Show

Again I worked on the laserdisc screen, setting the opencard script to see if there is a number entered within the field storing the laserdisc numbers, and if there is not, to put the word "none" into the field. I tried it using " " to put a space into the line, and also tried putting empty into the line, but I was unable to get the slide show button to read the space or the empty field so that it could skip those lines.

The Print Feature

I made my first attempt at writing the scripts for printing. It took a few tries before the printing of the text came out at all close to what I had envisioned. Although I can get the cards to print without any difficulty, it is a disappointment, because the picture windows do not print, leaving a blank space where the image is located on the screen.

Thesis Show Day

April 24, 1992

The day that I had dreaded and yet at the same time looked forward to with anticipation for so long had now arrived. Before the set up, I created a presentation on William Golden using the images that I had previously scanned and the book, Nine Pioneers in American Graphic Design, as the source for the text. Because the text had been prepared in advance, the entire process of creating the presentation of 9 screens took less than an hour.

At three in the afternoon, I, with the assistance of two of my friends, my husband, and one of my sons, carried down the computer table and computer hardware. The next problem was getting the plugs to fit and to reach. That took more time than carrying down the equipment. There was no electrical outlet near the space that had been given to me. The closest place on the floor had one outlet with already three cords plugged into it. Then we ran into problems with three prongs on the extension cord and the outlet having only spaces for two prongs. It took a while but finally we managed running an orange extension cord around the floor to make a connection.

Once the machines were plugged in and the cords as concealed as we could get them, I loaded the newest version of PresentationMaker onto the hard drive, along with the William Golden presentation, and all of the picture files. As I opened the program, the introduction ran fine, and the initial part of PresentationMaker was fine. When I looked at the help card for the created stacks, I noticed that the fonts were not coming in correctly, and in fact, ran off the fields. This was very disappointing to me as I had installed the Univers font that should be being read by that field. I took a moment and unlocked each of the fields, and changed the text by either editing it quickly or deleting some of the spacing until I got it to fit onto the page. Everything else appeared for the moment anyway to be working fine, so I shut the machine down, and went home to get ready.

Thesis Show

It was 6:45 when we arrived back at RIT for the thesis show. On walking in the door, I was greeted by two of my former teachers from Jamestown Community College; it was great to see them. I showed them my project, and they marvelled, mentioning that HyperCard was not a program with which they had had any success. One of them taught Art History courses, and felt that the PresentationMaker might be very helpful.

By 7:00 there were any number of people arriving, so that I was continually showing and explaining the project. The introduction was eye catching, and many people just stopped to look at that. The steady pace of people kept on, so that there was barely a moment to turn around until all of a sudden some of the lights were turned off and I realized that it was 9:00.

Of the various interested people during that time there were a number of professors who were interested, and also a person in the nursing profession who had to give lectures. They were the most interested as it had personal application for them. Professor Hodik sat down and actually tried it; the most difficult part for her was adjusting to the movement of the mouse in terms of controlling the events on the screen. But overall she was delighted with the program and the resulting presentation. She was most thrilled about the fact that at a click of a button, the font changed in front of her eyes, or an image could be pasted into the screen and moved around. Twice I was asked about selling the program, which was something that had never entered my mind, and I had no idea how to answer them.

Overall, the comments I received were most encouraging and supportive. At three different times students in the graphic design master's program stopped for a look, and their comments made reference to the sharp clean look, both within the program and the resulting presentations. Coming from them, I take that as a high compliment.

One of the nice things that happened for me was for the first time, I had to stand back a bit, and could take a good hard look at the program, not only through my eyes, but also through the eyes of those around me. I think that for the last few weeks I have been so engrossed within the process of creating it, and developing the scripts and design of it all, that I hadn't really seen the entirety of it all. I will have to say I was pleased; it was a difficult project and it has come a long way. For the most part the scripts and programming are working solidly and reliably.

However, as I observed others using the program, I did spot several little glitches. While they did not cause any major problems, I made mental notes so that I could remember and adjust the scripts. On layout #4, as the title field was moved around, when moved back to its highest location, it went up behind the graphic, and it should have lined up just below the graphic. On showing the notes field, I realized that the original notes field was still in place, where I matched the size of the notes to the size and location of the subtitle field. Clicking to rename a screen on the MoveCard caused an error message to appear "can't understand UpdateFldContents" that needs investigating.

April 25, 1992

I began by sitting down and making a list of the mistakes that I had noticed the evening before. Two other thoughts came to me: the introduction could use the line "Click anywhere to begin" on the screen. When the user is on a card on which the palette options are not appropriate, the wrong choice should be intercepted. This could be easily done by testing the background and informing the user that the choice is not presently available.

Refinements

The Phantom Scripts

There are two other problems that have shown up, and appear to be HyperCard phenomena, for I have not been able to eliminate them, much less understand them. One has been mentioned before. As a result of importing or moving a picture, the card script has to be rewritten. The card script is stored on a hidden field on the Help Card of the new stack. When the lines of the script are rewritten into that hidden field, those particular lines show up on the current card where the image is being imported or moved. If you would move to the next card and come back they won't be there. But they are there at the time, and the user might assume that the picture has to stay in that position just to cover up that text, which seems to appear from nowhere.

The second problem is more difficult to track down as it is a "sometimes" type of problem. Sometimes when a card is opened only the bottom half of the text in the title field will show. The title looks like it is half out of the field. If you click on the screen (if the background is black), the text jumps back onto its proper line. If the screen's background is white, then if you run the cursor through the text, selecting it, then the text moves back to its proper place. Perhaps because of the nature of the variety of font possibilities, I have not been able to get a clear picture of what exactly the cause is. It, too, is annoying.

Revisions

I went through each of the layouts, changing the size and location of the "TextEntry" field (which appears when the notes button is pressed) to a uniform size and location. Then I also did that for the screens that had been created in the William Golden presentation.

The next task I tackled was to try to eliminate the problem of the script appearing when an image is moved or imported. First, I moved the hidden field to the background, and rewrote the scripts to use a background field. There was no change. Then I made an opaque button in the background, and set the hilite to true, to cover it; there was still no improvement.

In playing with the picture, I noticed an error in the scripting of the arrow buttons for the image. When a picture is moved, the current location is entered and the script of the card rewritten. The error is in the omission of also rewriting the name of the picture file. If a picture is imported, the name and the location are written into the hidden field "cardScript." Then if the user moves to another card and decides that an image should be moved, what happens currently is that only the new location of the image is rewritten. This means that the field "CardScript" now has the new location of the present image, but also the name of the last imported image, and both will be rewritten to that card's script. The problem was simple to correct; just write the name of the image as well as the image's location into the new card script.

I reworked the script for invert as it took two clicks to make it work; now it works on a single click. Also I worked on the movement of the title field for the error in layout 4 where the title can move behind the graphic. The problem was a variable used earlier in the program development that had been eliminated, but still existed in that script. Once that was changed to use a variable that did exist, the script worked fine.

April 27, 1992

I began serious work on the documentation of the thesis project. I opened the stack in which I had the information on Review of the Literature, and in the back created a button that would export the text: first the title of each card, then the main text body, and finally the reference for that information. I opened the text file in Microsoft Word, and ran it through the SpellChecker. That alone took more time than one would think, but there was a volume of words, and also all the names in the references and the computer terminology (which it didn't know) slowed it down. I also checked on the spacing between the lines to set up for importing into a page layout program. In QuarkXPress, I created the master pages as the basis for the text, and then imported the text. I moved the titles into the column and made the size of the type of the references smaller, so that the main text stood apart. I then used the same layout for the first chapter, which is the Project Plan. Because of the length of the material in the Review of Literature, I spent most of the day on this part.

April 28, 1992

Professor Remington and I talked about the thesis show, and what was left to be completed. I mentioned that after working yesterday with the text from the Review of the Literature stack, I felt that we ought to give the user a chance to export the text to a text file, and I would like to add that option. I gave him a copy of the draft of chapters one and two.

When I met with Professor Ver Hague we also talked about the show and the comments and reactions to the project. We went over what had to be done to finish up the project, and I gave him a copy of the draft of chapters one and two.

I also showed him the problem of the script showing when an image is imported or moved. We tried adding to the script a line: "hide card field 'cardScript'", but that had no apparent effect.

I checked about the bindings for the thesis, and learned that if I was submitting all three copies to be bound, at that time, I would need to go to the Bursar's office and pay for it, bring a copy of the receipt, and all three would be sent in to be bound. As for turning them in, just have the sheets enclosed in an envelope.

May 2, 1992

I finished the connection for the laserdisc, so that a global controls whether the connection is set to on or off. This can be accessed through either the button on the laserdisc card or the "set laserdisc" button on the Presenter palette.

Since printing the card without the images was virtually useless, I changed that option to print the text to a file. Then I checked the programming, to make certain that options from the palettes were not accessible from the help cards is they were not appropriate.

May 1992

I am presently completing the documentation for the written portion of this thesis, and have had opportunity to demonstrate the program a number of times.

Missing Page

Chapter 4: Evaluation & Summary

This Master of Fine Arts thesis project, PresentationMaker, has proven to be a challenge, but extremely rewarding. I have gained knowledge and insights into designing and developing an effective interface. Incorporating research findings and fielding suggestions and input from faculty and other interested persons also was a significant part. In evaluating this experience the following questions come to mind. What processes proved to be valuable in the creation of PresentationMaker? Is the program functional? How completely were the goals and objectives that were initially developed met? Was the project worthwhile?

Processes

There were a number of processes that took place in the duration of this project. On an individual level I researched the area of multimedia and its many components first, then developed the application. This proved to be the right order as it allowed me to gather many ideas, and use them along with incorporating my classroom learnings.

Time lines and flow charts were helpful. Even though they proved to be sometimes unrealistic, they allowed ideas to be explored and kept the process moving. Daily journaling proved to be invaluable, both as a record keeping tool and in producing the documentation.

It was helpful for me to begin the thesis year with the concept for the project generally identified. This allowed me to move forward with preplanning and reflection before the actual implementation began.

Throughout the project I tried hard to pace myself, so that the work was spread more evenly over the period of time. This also allowed me time for reflection and review.

On arriving at a mental or computer deadlock, I would get away from the project and later come back to it fresh. This produced better results than constantly grinding away at a project, no matter what my level of exhaustion.

I was extremely fortunate to have professors that allowed me to have one to one sessions with them on a regular basis. This proved to keep the project on track, gave me fresh ideas, and served to evaluate the ideas and concepts in the project as it progressed.

In addition the group process (the weekly GDA meetings) was valuable and added additional feedback on a regular basis. While at times it proved frustrating, in the long run, the many comments helped me to evaluate the overall effectiveness of what I was doing. It allowed me to refocus or shift direction more easily than if feedback came less often.

Functionality

PresentationMaker has the potential to be a functional tool for an instructor or presenter. It would allow a person who has little or no knowledge of HyperCard or its programming language, HyperTalk, to create a presentation. It is user friendly allowing text and images to be combined easily.

Its multifaceted dimensions provide options for the user that require pages of scripting. Choices are made with the click of the mouse and the required scripting is completed for the user and inserted into the stack or the card, as needed.

Completeness

The goals and objectives that I initially set up were either completely or in large measure met. The research and initial planning were done sufficiently. The actual development of the project was done with thorough documentation. As a prototype this project was completed. However, PresentationMaker has not yet met the criteria of being a tested project.

It was probably not realistic in a one year project with other class work to have totally completed it. In order to finish the documentation before the end of the third quarter, the testing phase could not be done. Presently it needs substantial testing by potential or actual users, which would likely result in some modifications and corrections to its design and/or programming to make it a fully usable or marketable program.

Worth

Personally, I felt that PresentationMaker proved to be a most worthwhile project. It stretched and developed my skills as a programmer and interface designer. It provided a problem of large enough magnitude to be a desirable challenge, and while frustrating at times, proved to be rewarding when accomplished. It brought together skills and knowledge gained from many sources: computer graphics and AVI classes, research findings, professors, colleagues, and individual initiative and discovery. The integration of ideas and knowledge into one project that made its development exciting.

Although to this point there has been rather limited contact with potential users, those who have experimented with PresentationMaker seem to be extremely interested in the project and found it worthwhile. Also PresentationMaker contributes to the GDA project in that it is compatible and can make those images more readily usable in the classroom.

Appendix A: Time Schedules

Proposed Timeline #1

Week	Date	Tasks
16	Dec.2-5	Clarify project ideas Time Line
15	Dec. 9	Begin FlowChart Continue Research
14	Dec. 16	Finish Research Work on Overall Layout
13	Jan 6	First Draft of Chapters 1 & 2 Work on importing Text
12	Jan 13	Chapter1 & 2 Revisions Gather data & images (Agha & Golden)
11	Jan 20	Writeup of Journal to date Work on importing images (b/w bitmaps)
10	Jan 27	Revisions to overall layout, flowchart, etc. Work on importing color images
9	Feb 3	Continue Revisions Work on output methods
8	Feb 10	Write up of Journal to date Work on introduction and help
7	Feb 17	First major testing Revisions
6	Mar 9	Continue Revisions Write up of Journal to date
5	Mar 16	2nd major testing Begin other written parts of thesis
4	Mar 23	Revisions to program Continue writing
3	Mar 30	Final Evaluations and Recommendations Draft of Final Documentation
2	Apr 6	Last revisions Create final documentation
1	Apr 13	Prepare materials for show Documentation to printer

Timeline #2	Week	Date	Tasks
	16	Dec.2-5	Clarify project ideas Time Line
	15	Dec. 9	Begin Flow Chart Continue Research
	14	Dec. 16	Finish Research Work on Overall Layout
	13	Jan 6	First Draft of Chapters 1 & 2 Templates: design & control in HyperCd
	12	Jan 13	Chap.1 & 2 Revisions Gather data & images (Agha & Golden)
	11	Jan 20	Write up of Journal to date Work on importing images (b/w bitmaps)
	10	Jan 27	Revisions to overall layout, flowchart, etc. Work on importing color images
	9	Feb 3	Continue Revisions Work on output methods
	8	Feb 10	Write up of Journal to date Work on introduction and help
	7	Feb 17	First major testing Revisions
	6	Mar 9	Continue Revisions Write up of Journal to date
	5	Mar 16	2nd major testing Begin other written parts of thesis
	4	Mar 23	Revisions to program Continue writing
	3	Mar 30	Final Evaluations and Recommendations Draft of Final Documentation
	2	Apr 6	Last revisions Create final documentation
	1	Apr 13	Prepare materials for show Documentation to printer

To date I have accomplished the following:

- made an in depth analysis of the situation and problem
- outlined goals and objectives for the project
- studied over 40 articles/books on interactivity and multimedia projects
- completed a HyperCard summary of the research that has been done
- met regularly with Roger R. and Jim V.H.
- met occasionally with Barbara H. and Mark C.
- created a time schedule including major objectives to be accomplished
- developed flowchart of overview for entire project
- drafted and revised flowchart of the input section of the program
- began first pass at input section of program:
 - created sample templates, trying out use of color resources
 - created palette containing imported color
 - worked at overall layout and contents of this section
 - experimented with the use of add-on menus as an option
- worked on button to import and place pict files
- presented regular reports to the GDA committee
- kept a record of my progress, problems and meetings

To date I would also have liked to accomplish the following:

- Finished the revisions of ch. 1 (Proposal) and 2 (Summary of Research)
- Gathered data and images on Agha and Golden
- Have the journal written up nicely to date
- Have the first part of the program completed and be well into the Editor part

In addition to these last I also must do the following:

- develop the 2nd part (Editor) and 3rd part (Output)
- create the introduction (using MacroMind)
- do thorough testing of the program
- continue revisions and writing of journal and reports
- complete evaluations

I have not made the progress that I had hoped, but I had set up a rather ambitious time schedule. This next quarter I should be able to turn my entire attention to progressing on this project and I am looking forward to that.

Written Work:

- ✓ Clarify project ideas
- ✓ Time Line
- ✓ Flow Chart
- ✓ Research
- ✓ First Draft of Chapter 1: Proposal/Plan for Thesis
- First Draft of Chapter 2: Summation of Research
- Chap. 1 & 2 Revisions
- ✓ Write up of Journal to date
- Revisions to overall layout, flowchart, etc.
- Final Evaluations and Recommendations
- Draft of Final Documentation

Project Work:

- ✓ Begin Template Designer; expand flowchart
- ✓ Create 5 templates
- ✓ Design and develop functionality of 5 palettes:
 - ✓ 1. to browse and choose template (Chooser)
 - load/save work
 - ✓ - 3 choices for graphic elements
 - ✓ - view in reverse; view grid
 - ✓ - tracks user history & saves in file
 - ✓ 2. control of layout (Director)
 - on screen help system
 - ✓ - control of fields: font choices, placement (Title,BodyText)
- Add cleanup to reset work areas
- Programming to separate Presentation from presentationMaker
- Create tools for Presentation program
- Connect interface to GDA laserdisc
- Gather data & images (Agha & Golden)
- Work on importing images (b/w bitmaps)
- Work on importing color images
- Work on output methods
- Work on introductory sequence
- Develop on-screen helps
- First major testing and Revisions
- Begin documentation of PresentationMaker
- 2nd major testing and Revisions to program
- Refine documentation
- Last revisions and create final documentation
- Prepare materials for show
- Documentation to printer

Apr 24 Thesis Show

Written Work:

- ✓ Clarify project ideas
- ✓ Time Line
- ✓ Flow Chart
- ✓ Research
- ✓ First Draft of Chapter 1: Proposal/Plan for Thesis
- First Draft of Chapter 2: Summation of Research
- Revisions: ✓Chap.1 & Chap. 2
- ✓ Write up of Journal to date
- ✓ Revisions to overall layout, flowchart, etc.
- Final Evaluations and Recommendations
- Draft of Final Documentation

Project Work:

- ✓ Begin Template Designer; expand flowchart
 - ✓ Create 5 templates
 - ✓ Design and develop functionality of 5 palettes:
 - ✓ 1. to browse and choose template (Chooser)
 - ✓ - 3 choices for graphic elements
 - ✓ - view in reverse; view grid
 - ✓ - tracks user history & saves in file
 - ✓ 2. control of layout (Director)
 - ✓ - control of fields: font choices, placement (Title,BodyText)
 - ✓ Add cleanup to reset work areas
 - 3-31* Revise template8 to be choice of template only and store resources
 - 3-31 Work on script for new Choose button
(to create new stack, install palettes, scripts, & graphic element)
 - 3-31 Revise existing palettes :Director, Title, BodyText
 - 3-31 Create 2 new palettes: (palette, graphics, & scripting)
 - Image Importer - Work on importing b/w bitmaps & color images
 - View Screens - Work on output methods
 - 4-7 Connect interface to GDA laserdisc
 - 4-14 Develop on screen help system
 - 4-14 Gather data & images (Agha & Golden) for sample presentation
 - 4-14 First major testing and Revisions
 - 4-14 Begin documentation of PresentationMaker
 - 4-21 2nd major testing and Revisions to program
 - 4-21 Refine documentation
 - Prepare materials for show
- Apr 24 Thesis Show**
- Last revisions and create final documentation
 - Documentation to printer

* By this date this part was attempted with hopes of having it completed.

Written Work:

- ✓ Clarify project ideas
- ✓ Time Line
- ✓ Flow Chart
- ✓ Research
- ✓ First Draft of Chapter 1: Proposal/Plan for Thesis
- First Draft of Chapter 2: Summation of Research
- Revisions: ✓Chap.1 & Chap. 2
- ✓ Write up of Journal to date
- ✓ Revisions to overall layout, flowchart, etc.
- Final Evaluations and Recommendations
- Draft of Final Documentation

Project Work:

- ✓ Begin Template Designer; expand flowchart
- ✓ Create 5 templates
- ✓ Design and develop functionality of 5 palettes:
 - ✓ 1. to browse and choose template (Chooser)
 - ✓ - 3 choices for graphic elements
 - ✓ - view in reverse; view grid
 - ✓ - tracks user history & saves in file
 - ✓ 2. control of layout (Director)
 - ✓ - control of fields: font choices, placement (Title,BodyText)
- ✓ Add cleanup to reset work areas
- ✓ Revise template8 to be choice of template only and store resources
- ✓ Work on script for new Choose button
 - (to create new stack, install palettes, scripts, & graphic element)
- ✓ Revise existing palettes :Director, Title, BodyText
- ✓ Create 2 new palettes: (palette, graphics, & scripting)
 - ✓ Image Importer - Work on importing b/w bitmaps & color images
 - ✓ View Screens - Work on output methods
- ✓ Connect interface to GDA laserdisc
- ✓ Develop on screen help system
- ✓ Gather data & images (Agha & Golden) for sample presentation
- 4-14 First major testing and Revisions
- 4-14 Begin documentation of PresentationMaker
- 4-21 2nd major testing and Revisions to program
- 4-21 Refine documentation
- ✓ Prepare materials for show

Apr 24 Thesis Show

Last revisions and create final documentation
Documentation to printer

* By this date this part was attempted with hopes of having it completed.

Tasks and Progress to Date:

Jane Ann Settergren
May 1, 1992

Written Work:

- ✓ Clarify project ideas
- ✓ Time Line
- ✓ Flow Chart
- ✓ Research
- ✓ First Draft of Chapter 1: Proposal/Plan for Thesis
- ✓ First Draft of Chapter 2: Summation of Research
- ✓ Revisions: ✓Chap.1 & ✓Chap. 2
- ✓ Write up of Journal to date
- ✓ Revisions to overall layout, flowchart, etc.
- Final Evaluations and Recommendations
- Draft of Final Documentation

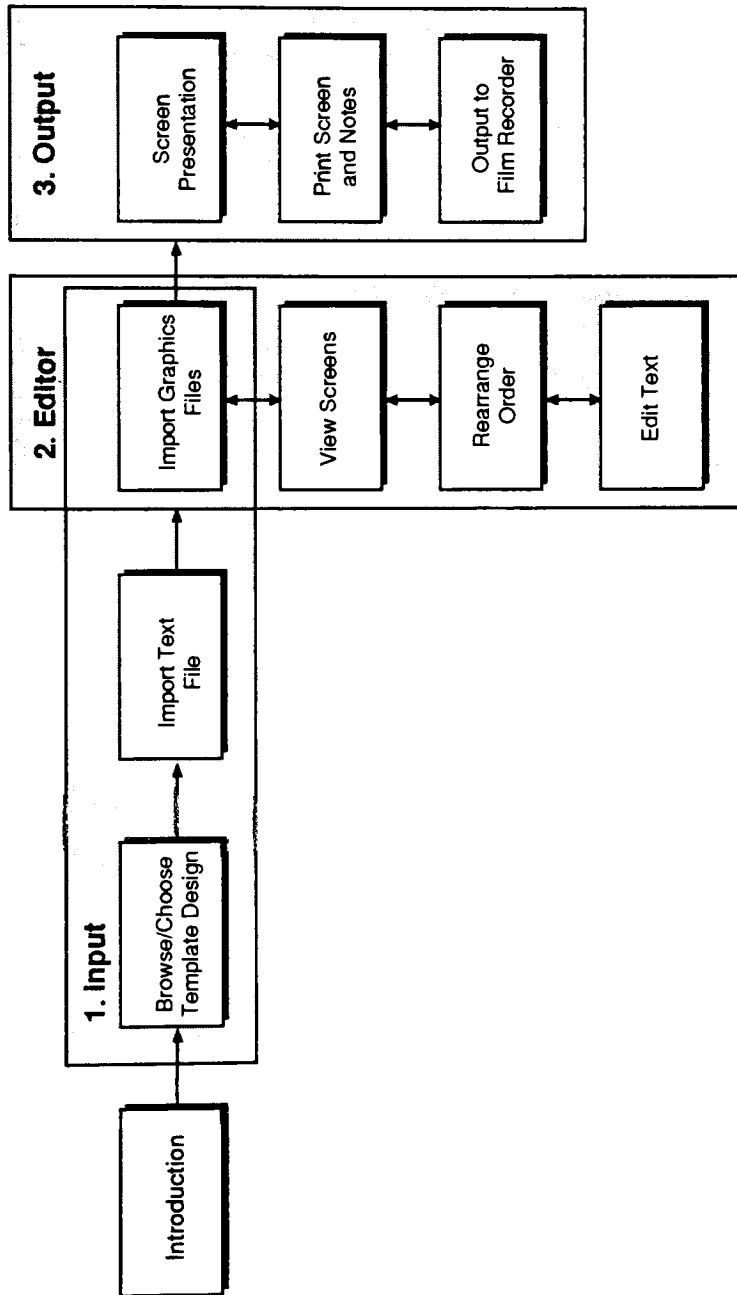
Project Work:

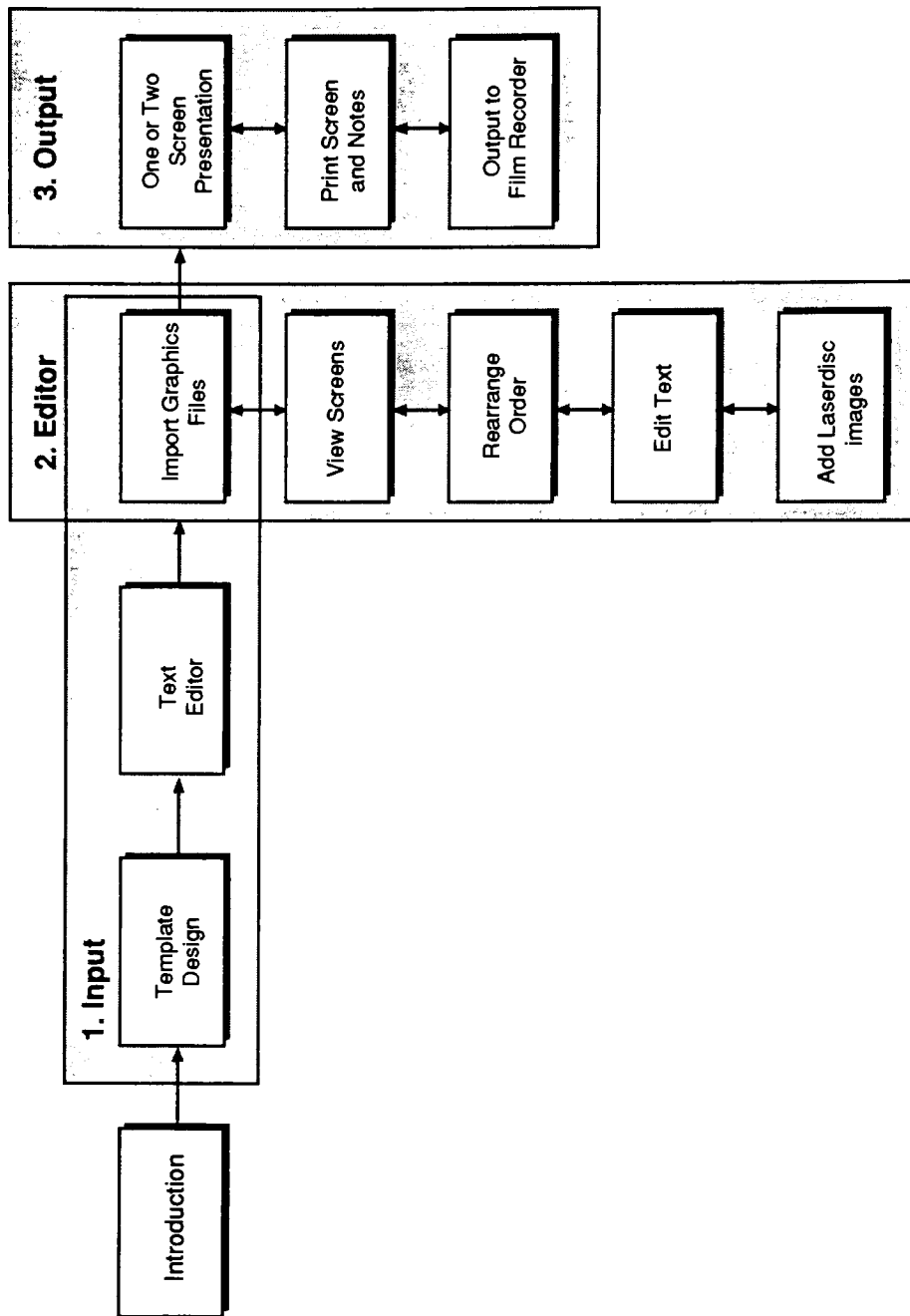
- ✓ Begin Template Designer; expand flowchart
- ✓ Create 5 templates
- ✓ Design and develop functionality of 5 palettes:
 - ✓ 1. to browse and choose template (Chooser)
 - ✓ - 3 choices for graphic elements
 - ✓ - view in reverse; view grid
 - ✓ - tracks user history & saves in file
 - ✓ 2. control of layout (Director)
 - ✓ - control of fields: font choices, placement (Title,BodyText)
- ✓ Add cleanup to reset work areas
- ✓ Revise template8 to be choice of template only and store resources
- ✓ Work on script for new Choose button
 - (to create new stack, install palettes, scripts, & graphic element)
- ✓ Revise existing palettes :Director, Title, BodyText
- ✓ Create 2 new palettes: (palette, graphics, & scripting)
 - ✓ Image Importer - Work on importing b/w bitmaps & color images
 - ✓ View Screens - Work on output methods
- ✓ Connect interface to GDA laserdisc
- ✓ Develop on screen help system
- ✓ Gather data & images (Agha & Golden) for sample presentation
- ✓ First major testing and Revisions
- ✓ Begin documentation of PresentationMaker
- 4-21 2nd major testing and Revisions to program
- 4-21 Refine documentation
- ✓ Prepare materials for show
- ✓ Thesis Show
 - Last revisions and create final documentation
 - Documentation to printer

* By this date this part was attempted with hopes of having it completed.

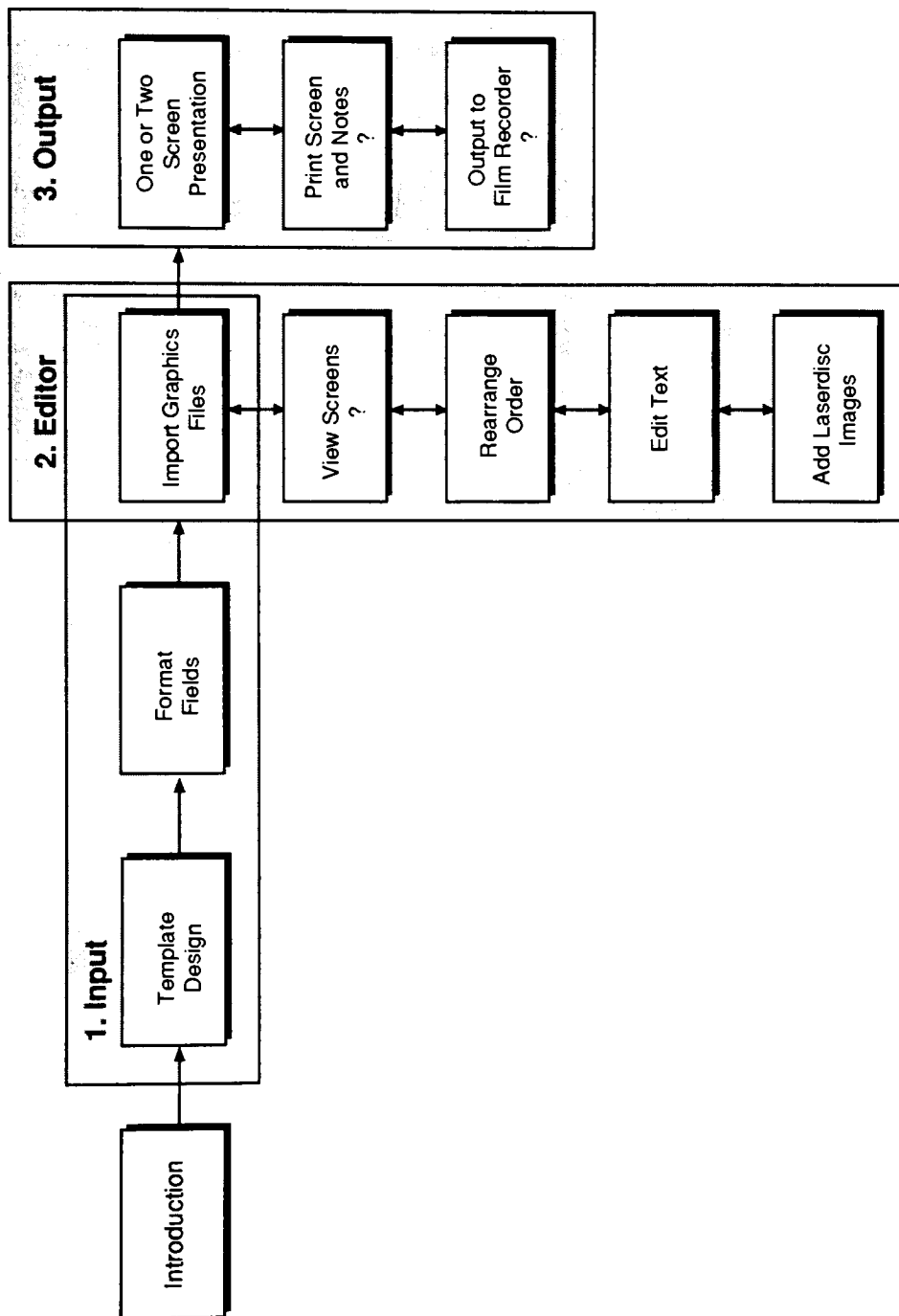
Appendix B: Flow Charts

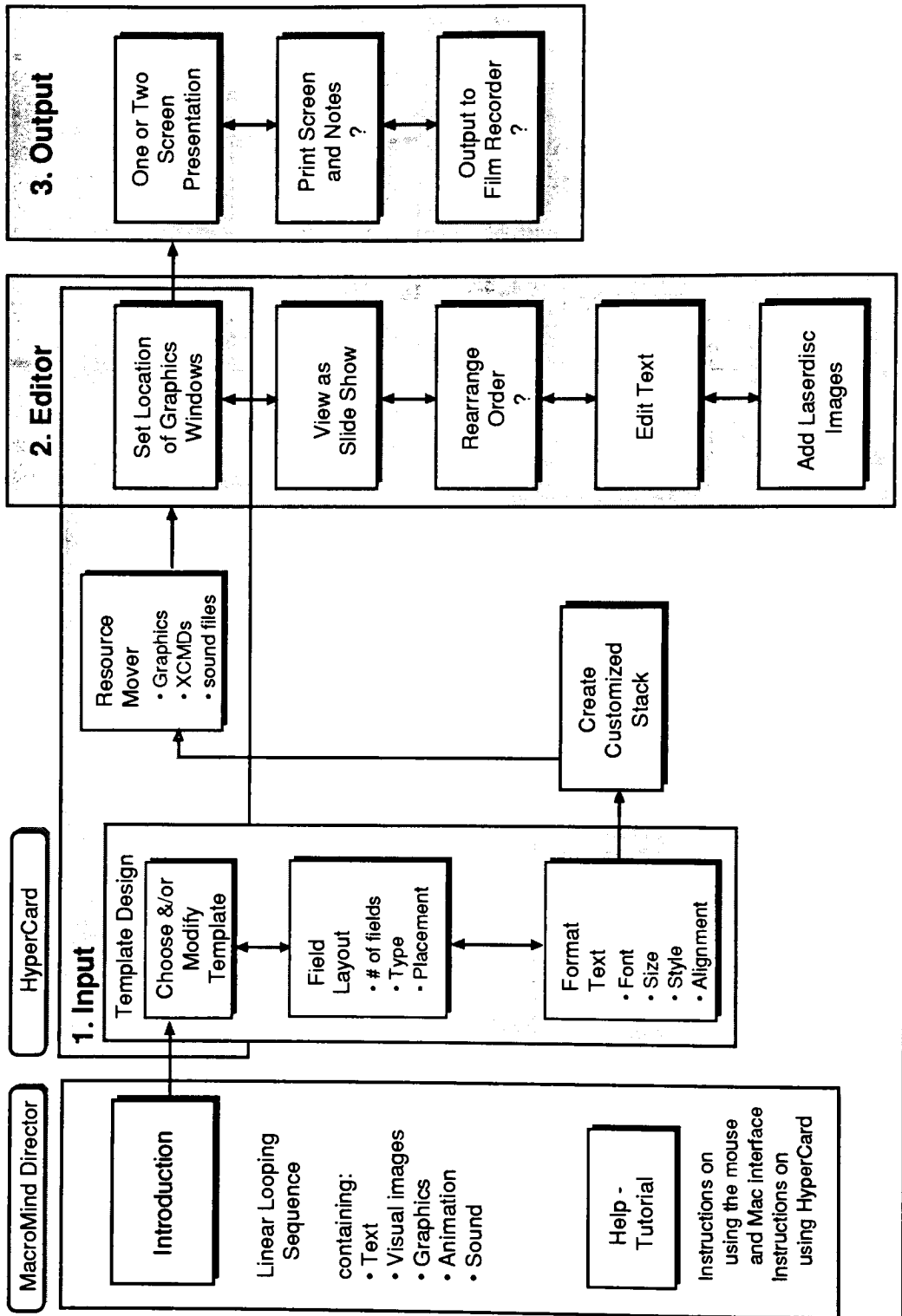
Jane Ann Settergren
PresentationMaker
12-11-91
1st Draft of FlowChart

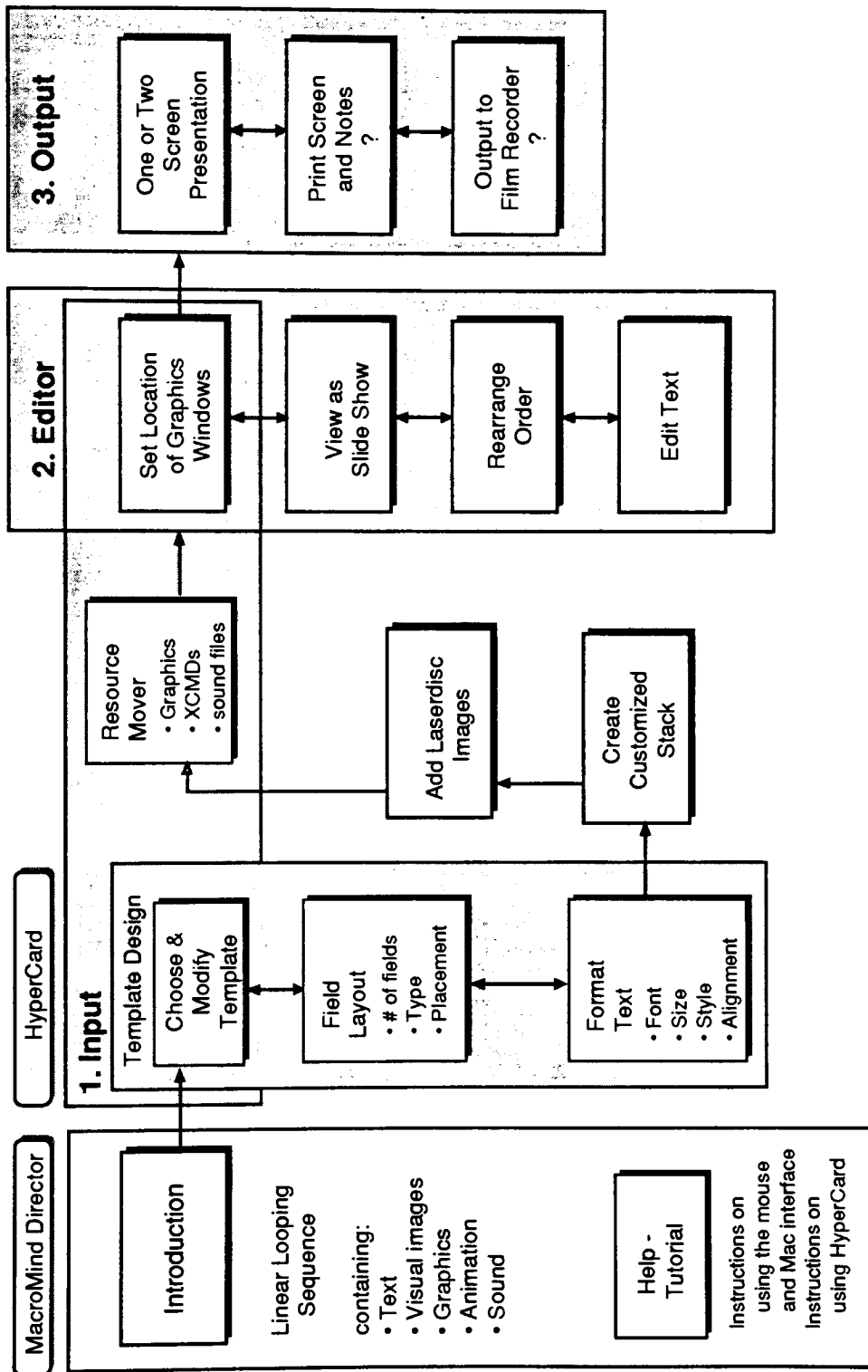


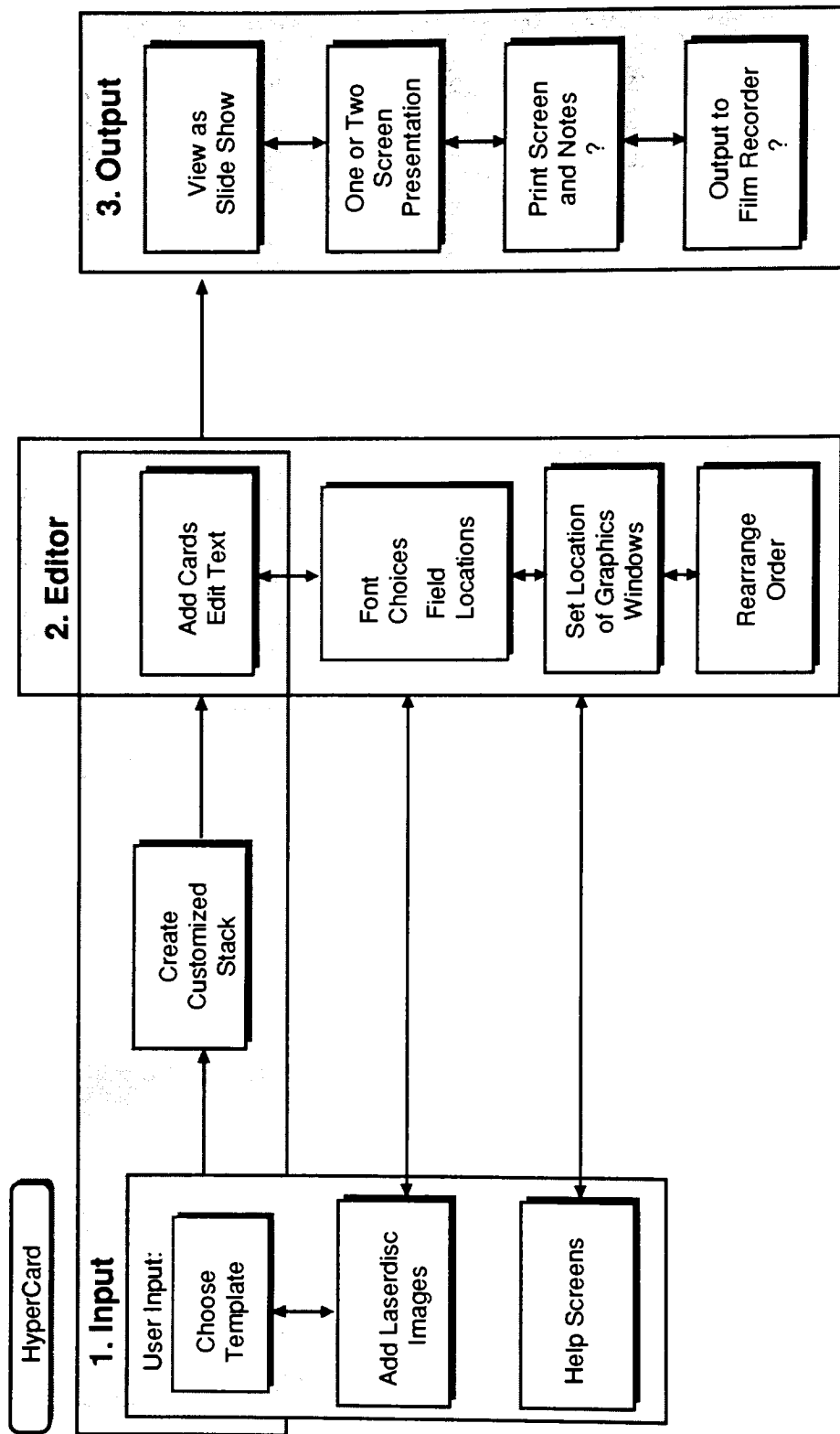


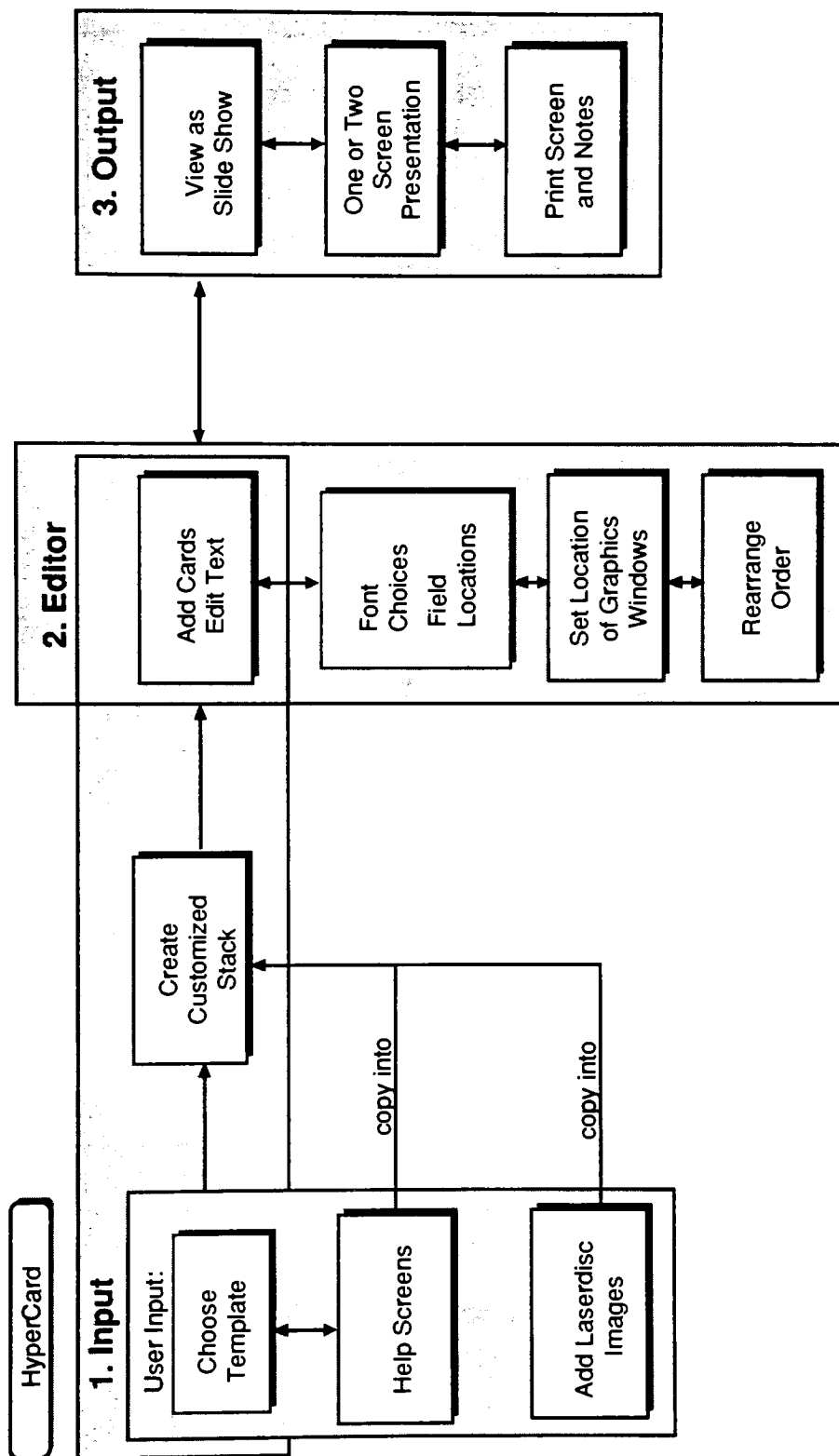
Jane Ann Settergren
PresentationMaker
1-6-92
3rd Draft of FlowChart





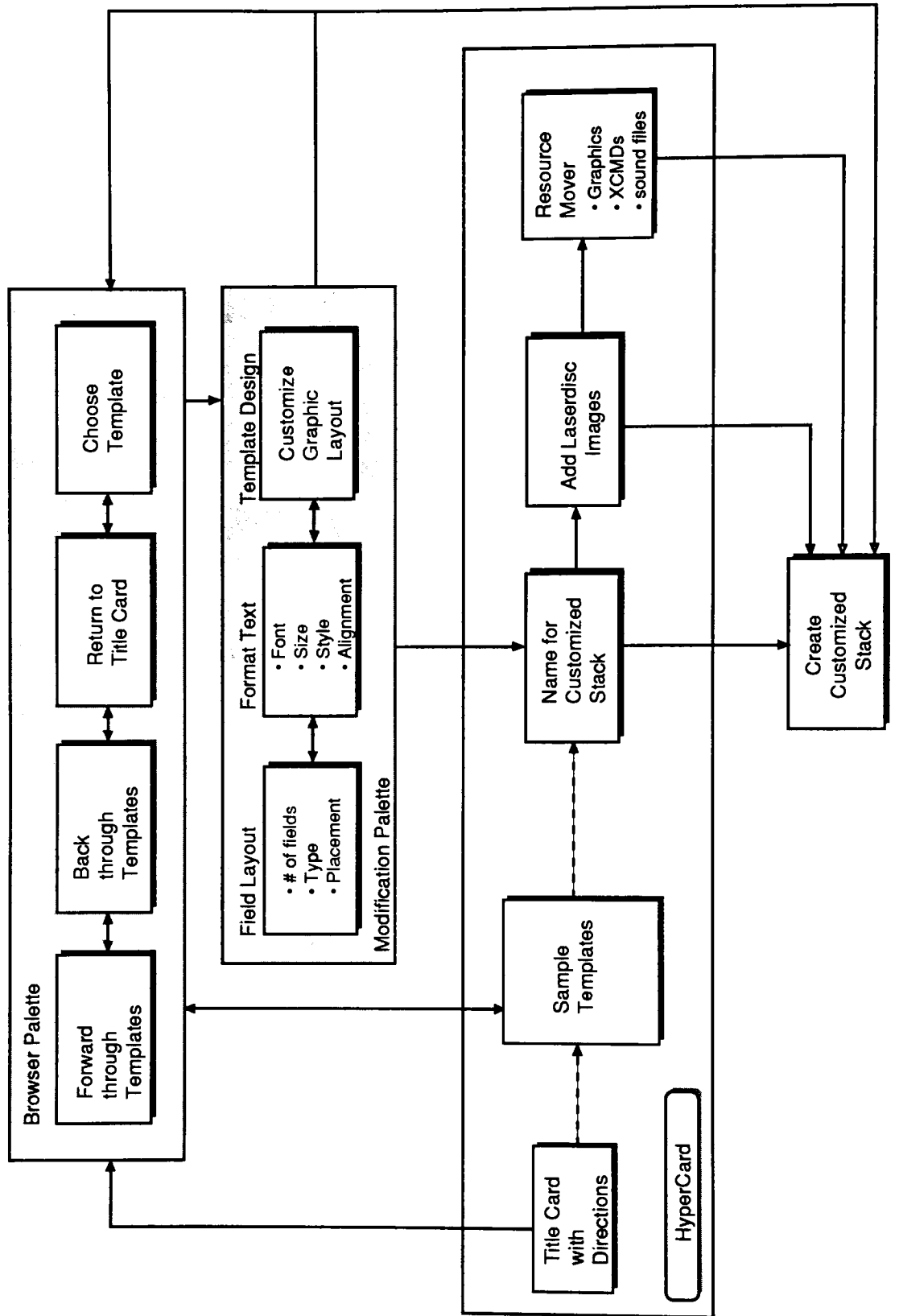






1. Input

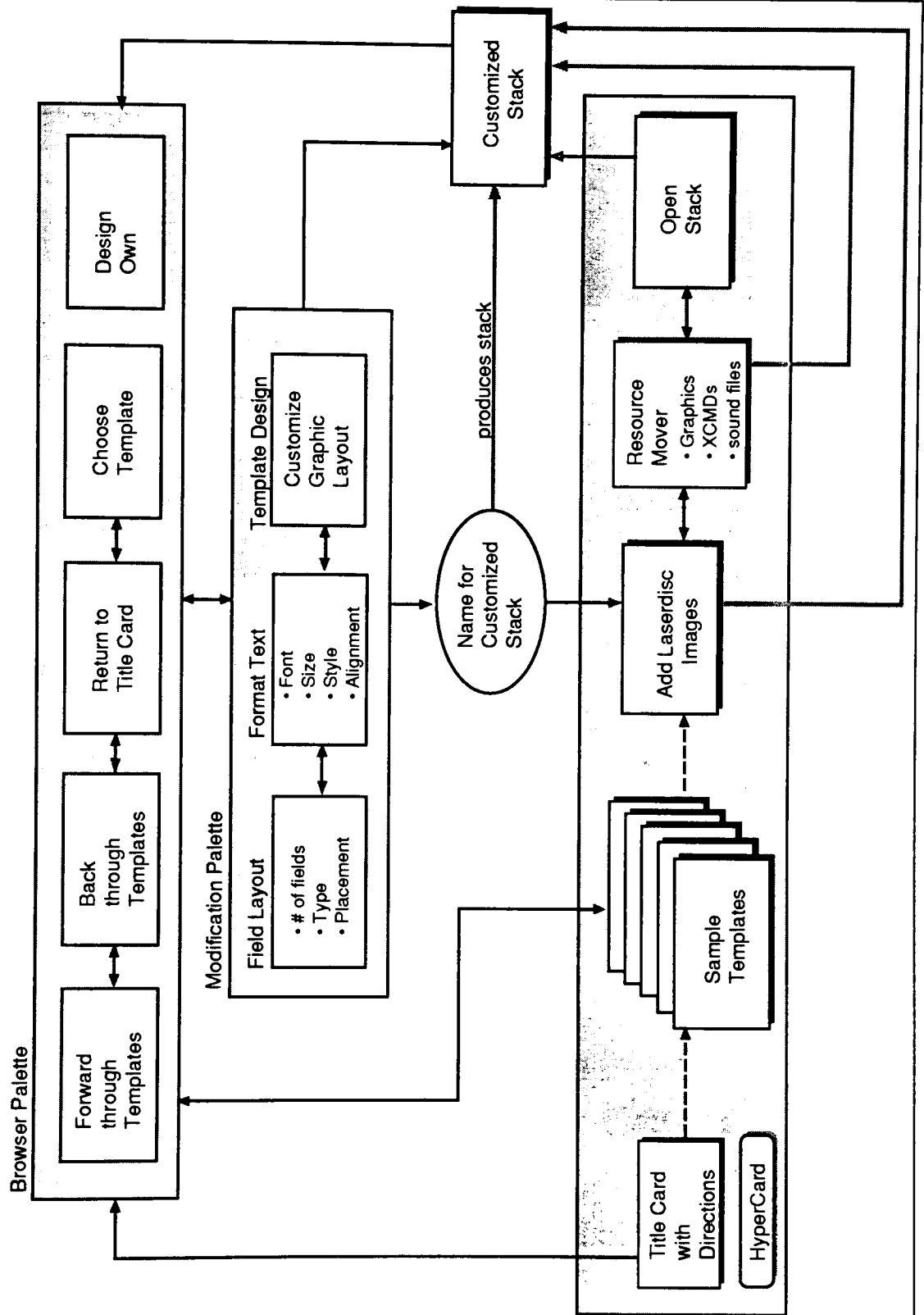
Template Design #1
Jan. 31, 1992



1. Input

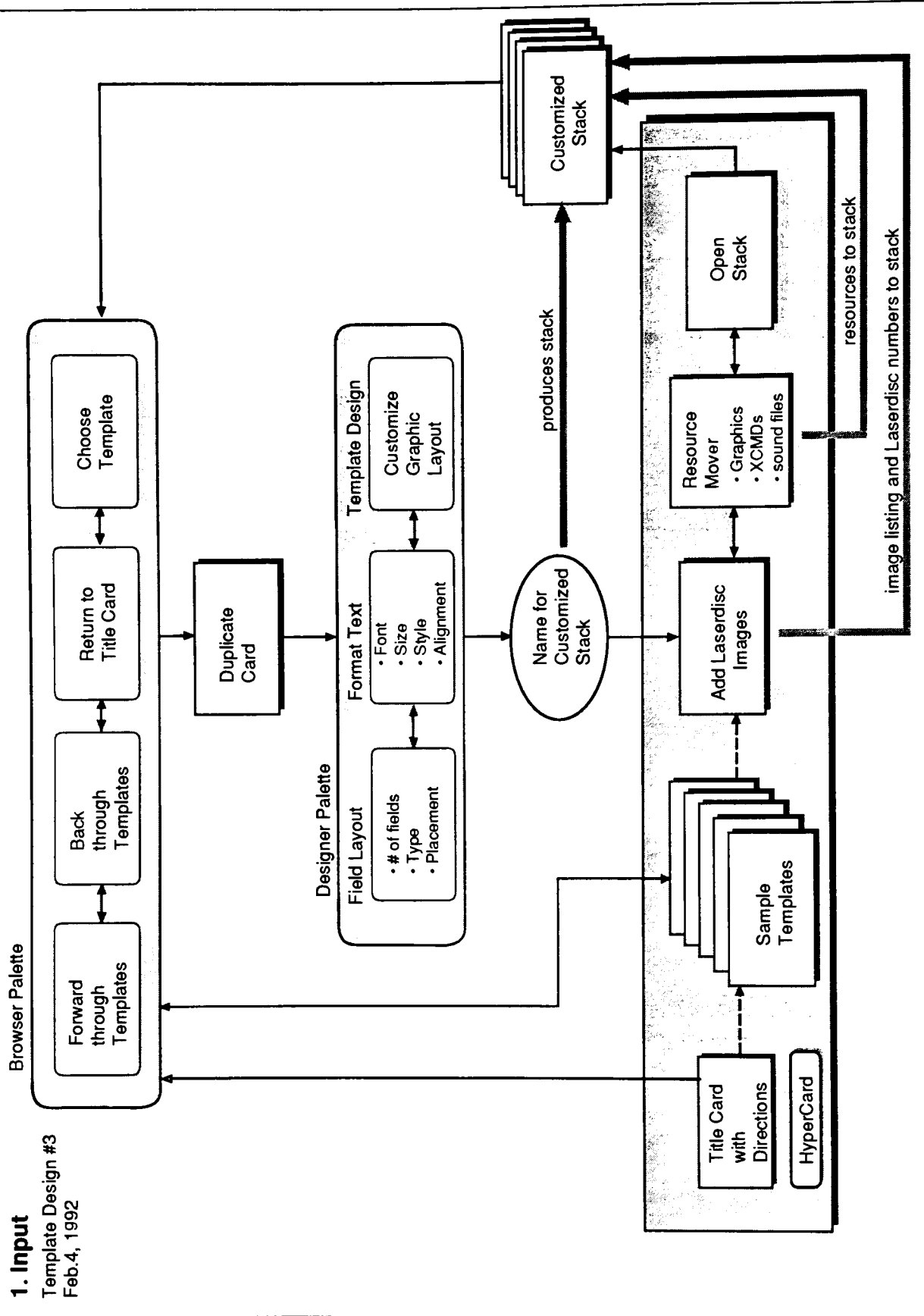
Template Design #2

Feb.1, 1992



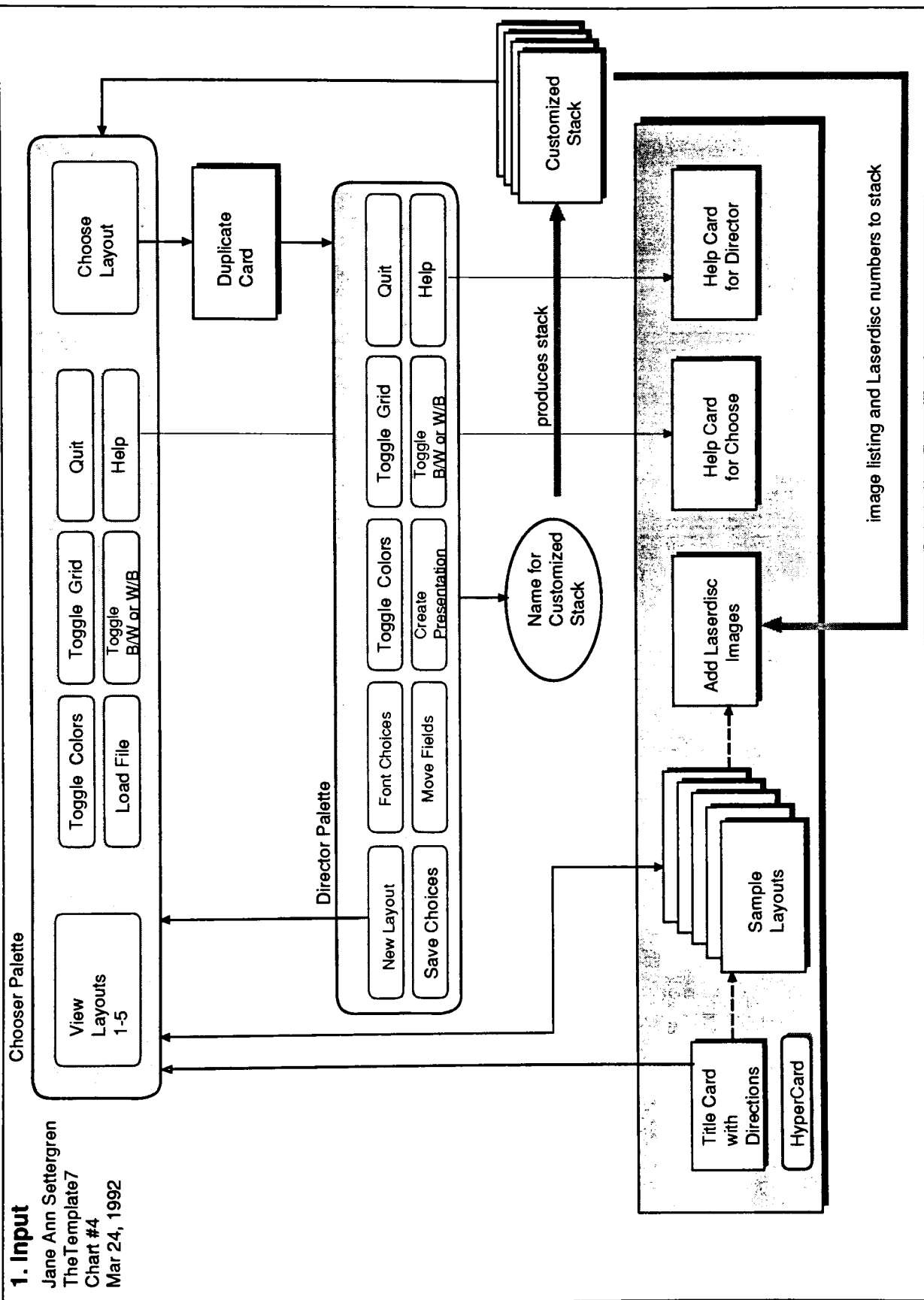
1. Input

Template Design #3
Feb.4, 1992



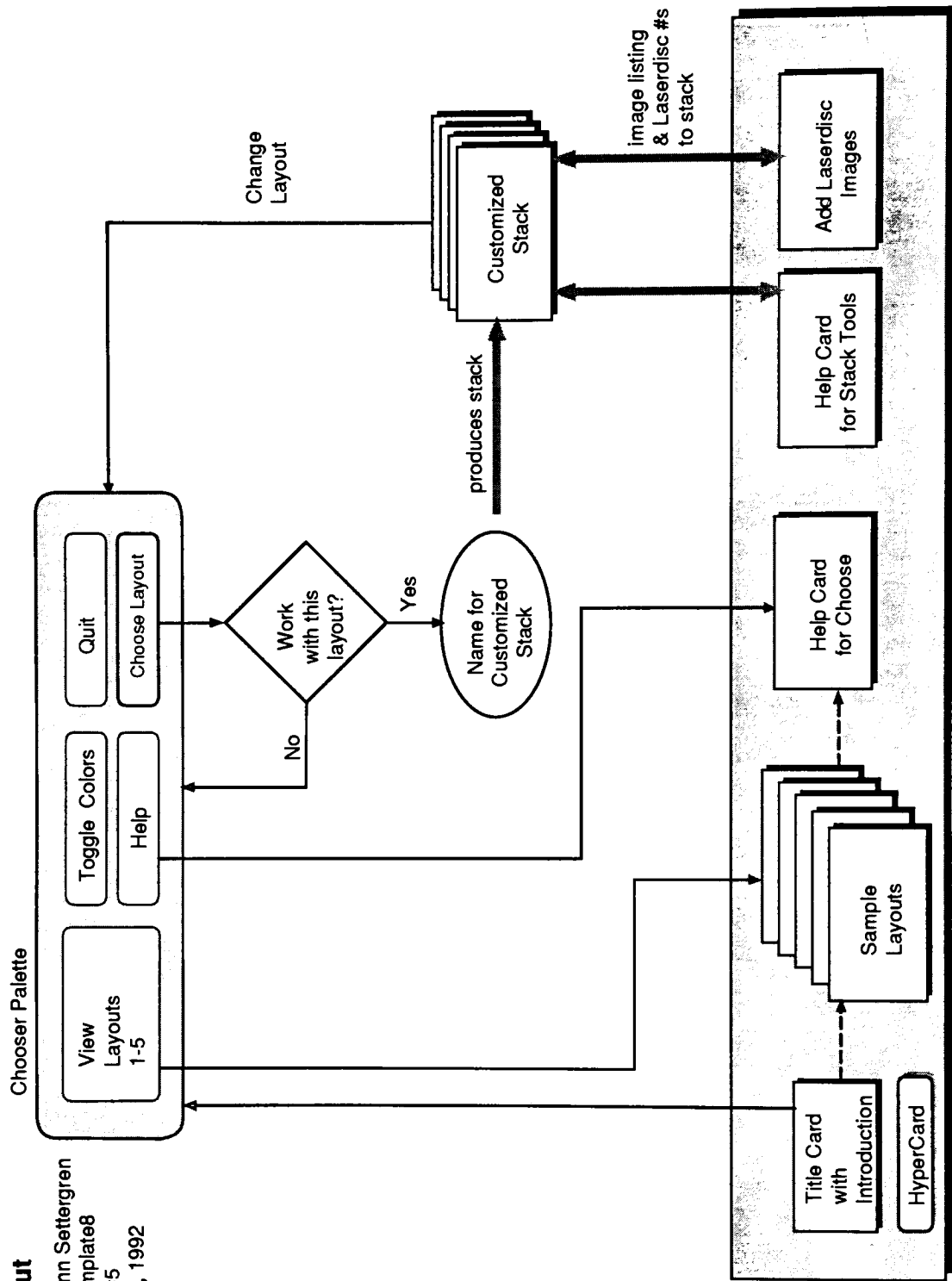
1. Input

Jane Ann Settergren
TheTemplate7
Chart #4
Mar 24, 1992

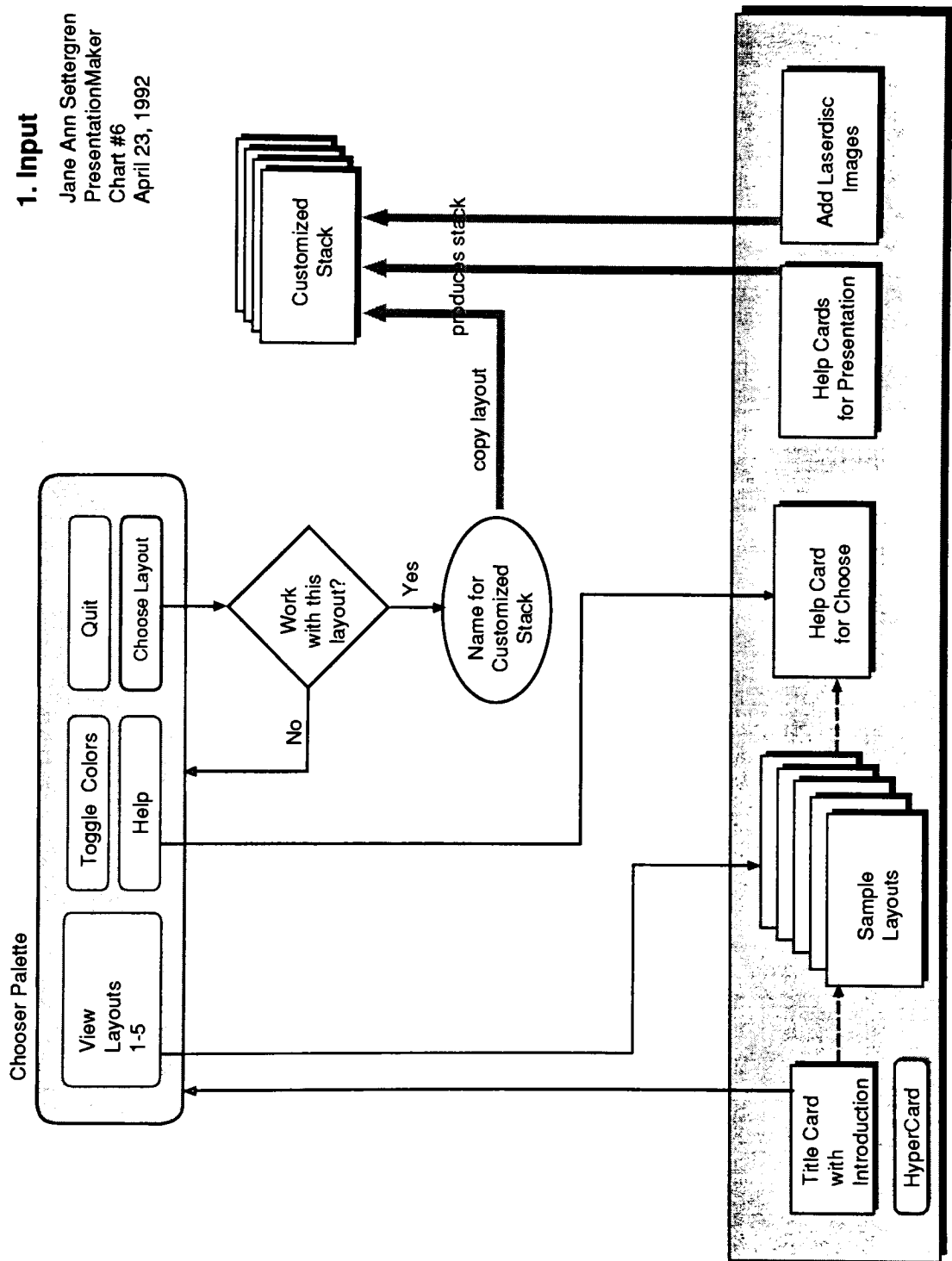


1. Input

Jane Ann Settergren
The Template8
Chart #5
Mar 25, 1992



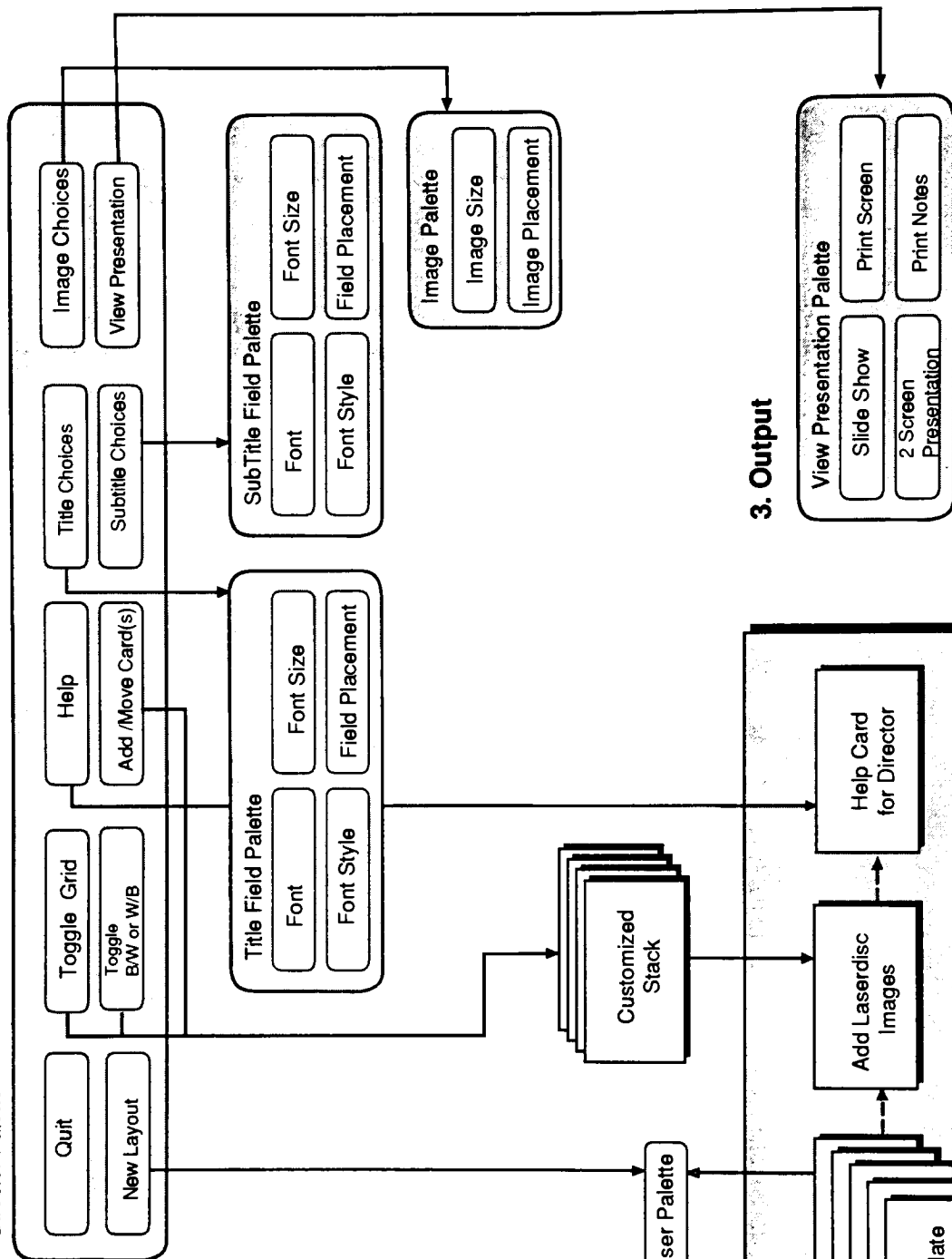
1. Input
Jane Ann Settergren
PresentationMaker
Chart #6
April 23, 1992



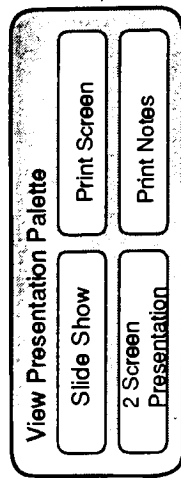
2. Editor

Jane Ann Settergren
Director1
Chart #1
Mar 25, 1992

Director Palette



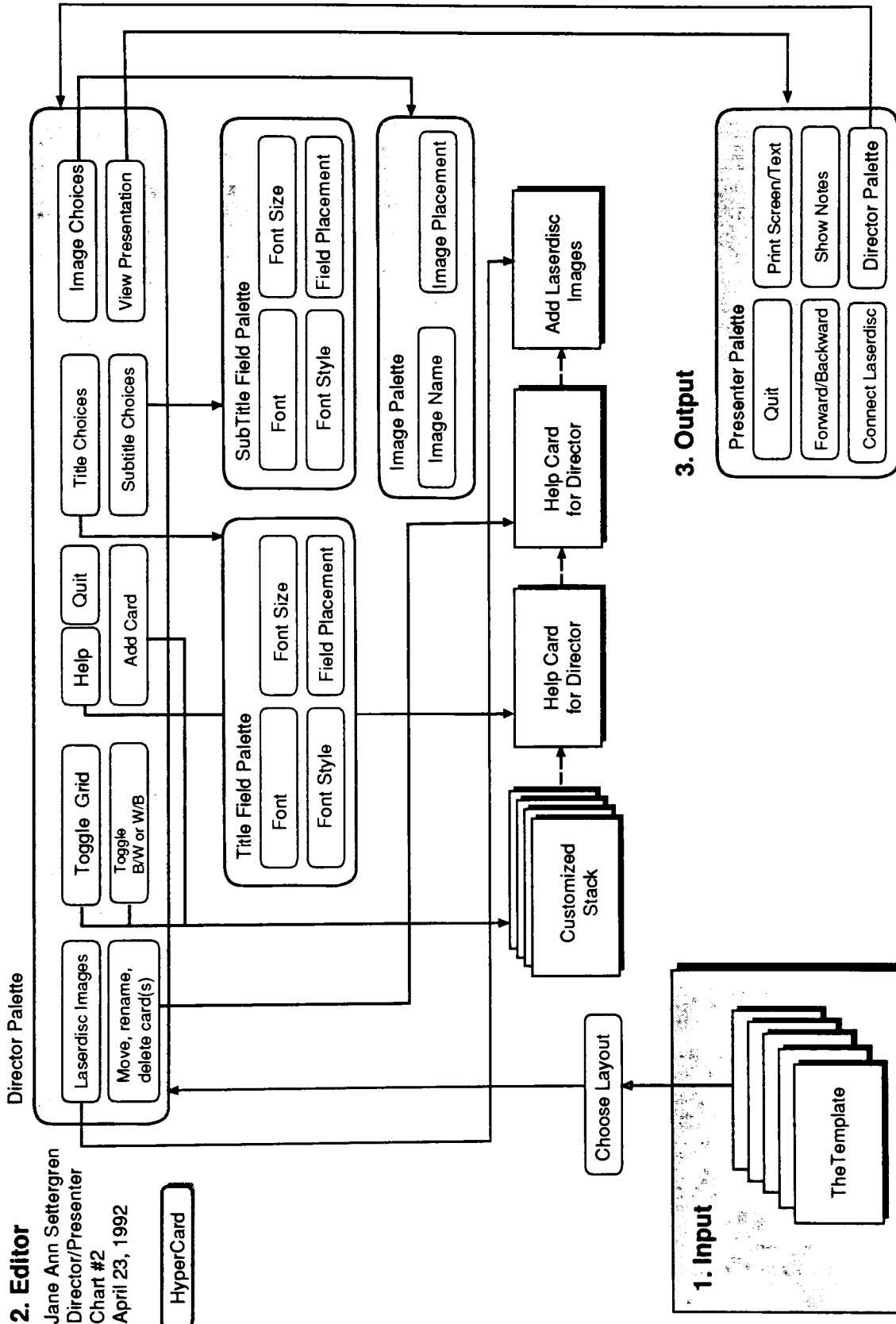
3. Output



2. Editor

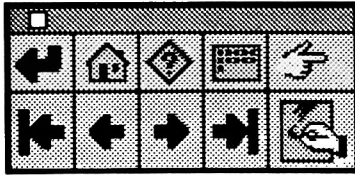
Jane Ann Settergren
Director/Presenter
Chart #2
April 23, 1992

HyperCard



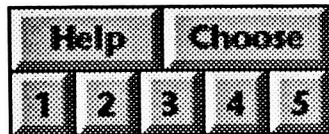
Appendix C: Palette Development

1. First Attempt at Colored Palette in HyperCard

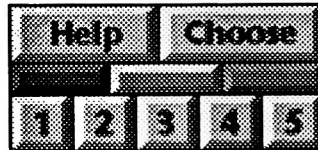


Experimenting
with Colored
Palettes

2. TheTemplate2

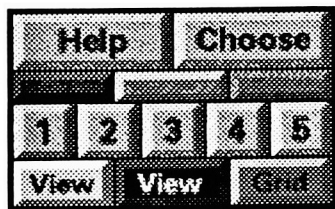


3. TheTemplate3

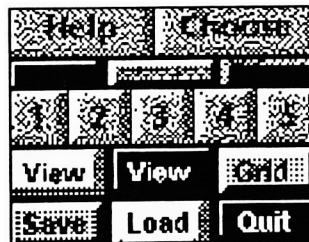


Development
of the
Chooser Palette

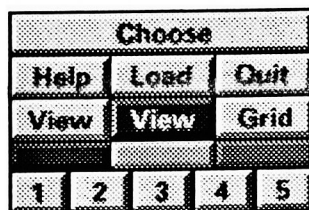
4. TheTemplate4



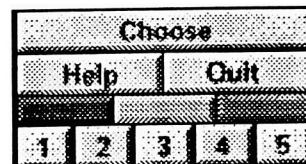
5. TheTemplate6



6. TheTemplate7

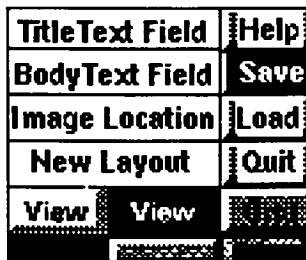


7. TheTemplate8



Development of the Director Palette

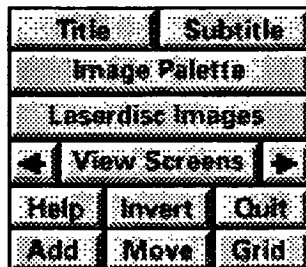
1. TheTemplate6



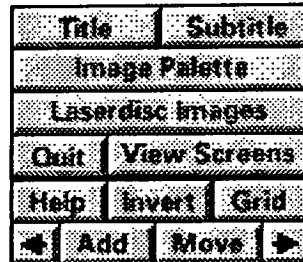
2. TheTemplate7



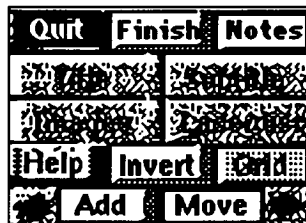
3. TheTemplate8.0



4. TheTemplate8.1



5. TheTemplate8.4



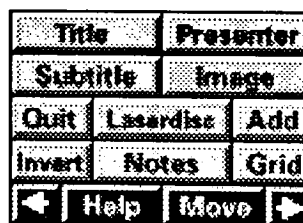
6. TheTemplate8.6



7. TheTemplate8.6 #2



8. The Template9.0



1. TheTemplate6

Helvetica	Plain	↑
Avant Garde	Bold	→
Times	Italic	←
Bookman	24 36	↓

2. TheTemplate7

Avant Garde	Plain		
Bookman	Bold		
Helvetica	24		
Times	36		
⬆	⬇	⬅	⬇

Development of the Title Palette

3. TheTemplate6

Helvetica	Plain
Avant Garde	Bold
Times	↑ →
Bookman	← ↓

4. TheTemplate7

Avant Garde	Plain		
Bookman	Bold		
Helvetica	18		
Times	24		
⬅	➡	⬅	⬇

Development of the Subtitle Palette

5. TheTemplate8.0

Import Image
↑ → ← ↓

Development of the Image Palette

**Development
of the
Presenter Palette**

1. TheTemplate8.4



2. TheTemplate8.8

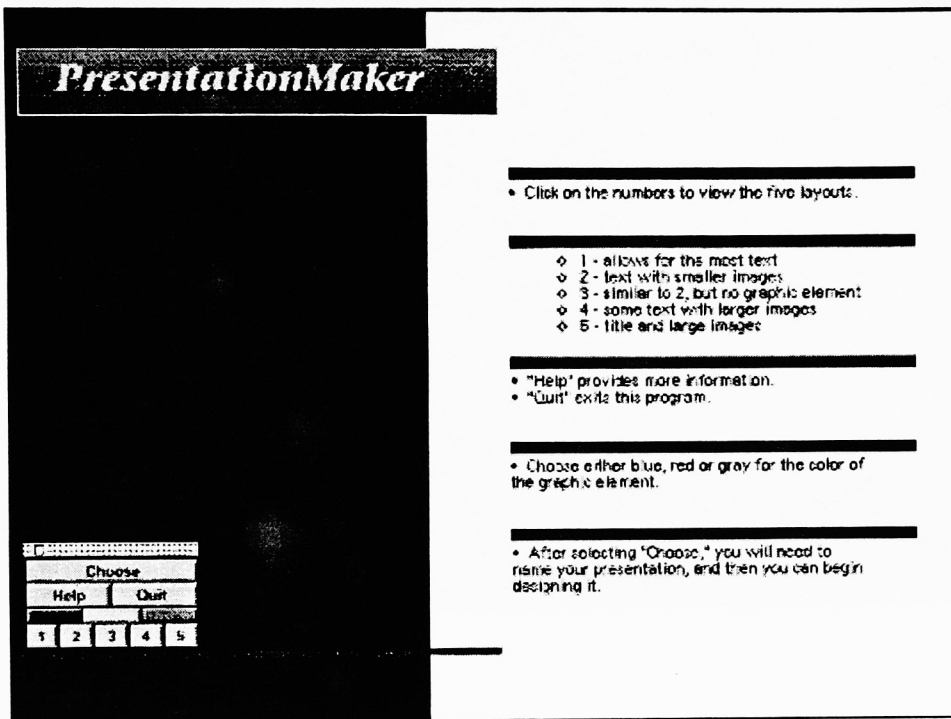


3. TheTemplate9

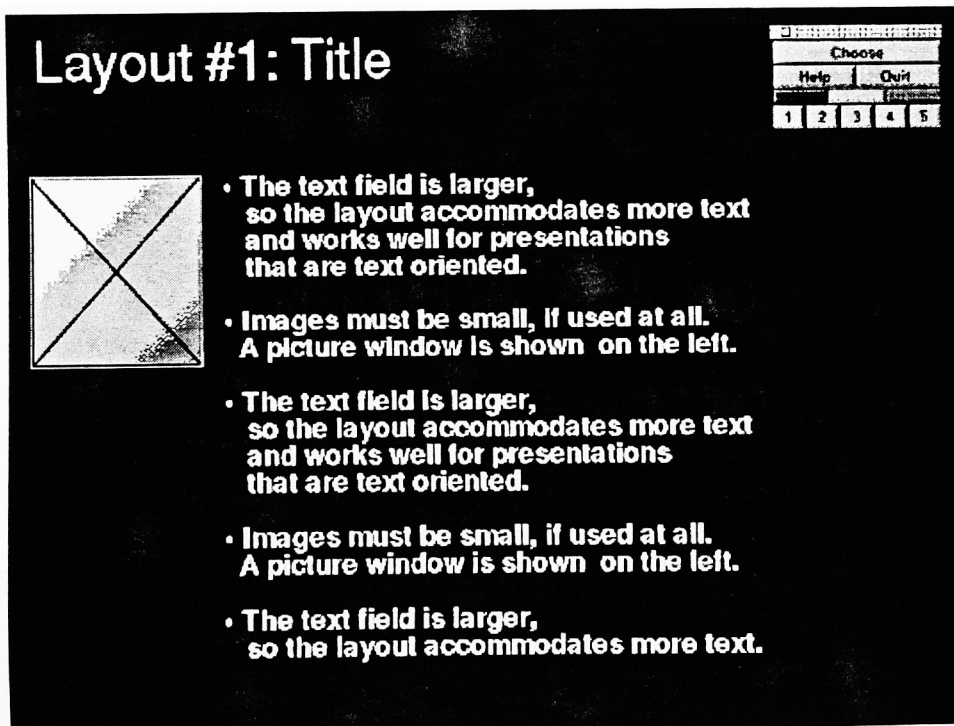


Appendix D: PresentationMaker Screens

Presentation Maker Title Card



Layout Choice #1



Layout Choice #2

Layout #2: Title

Choose

Help

Quit

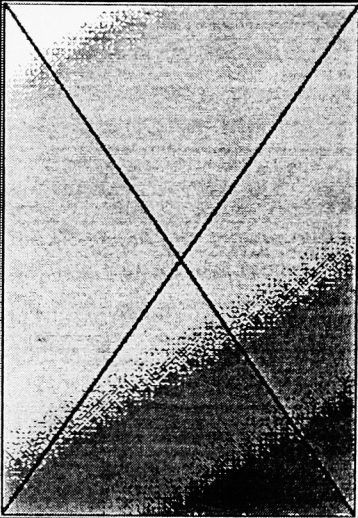
1

2

3

4

5



- Subtitle goes here: The text field here occupies about half the screen with room left for the addition of a medium-sized image.
- Subtitle goes here: The text field here occupies about half the screen with room left for the addition of a medium-sized image.
- Subtitle goes here: The text field here occupies about half the screen with room left for the addition of a medium-sized image.

Layout Choice #3

Layout #3: Title

Choose

Help

Quit

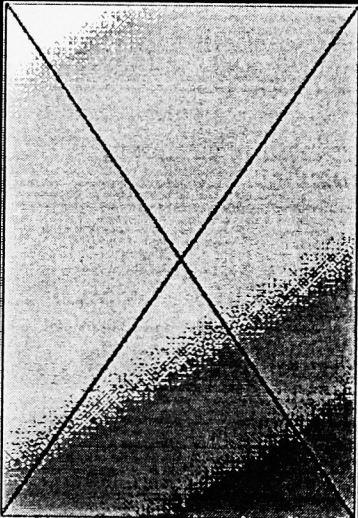
1

2

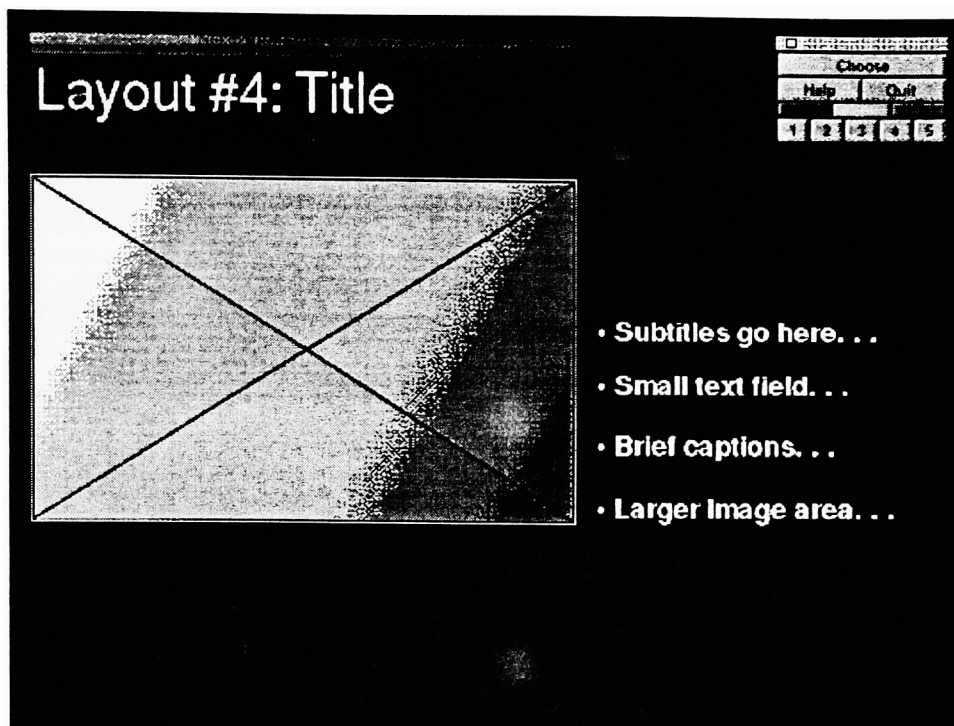
3

4

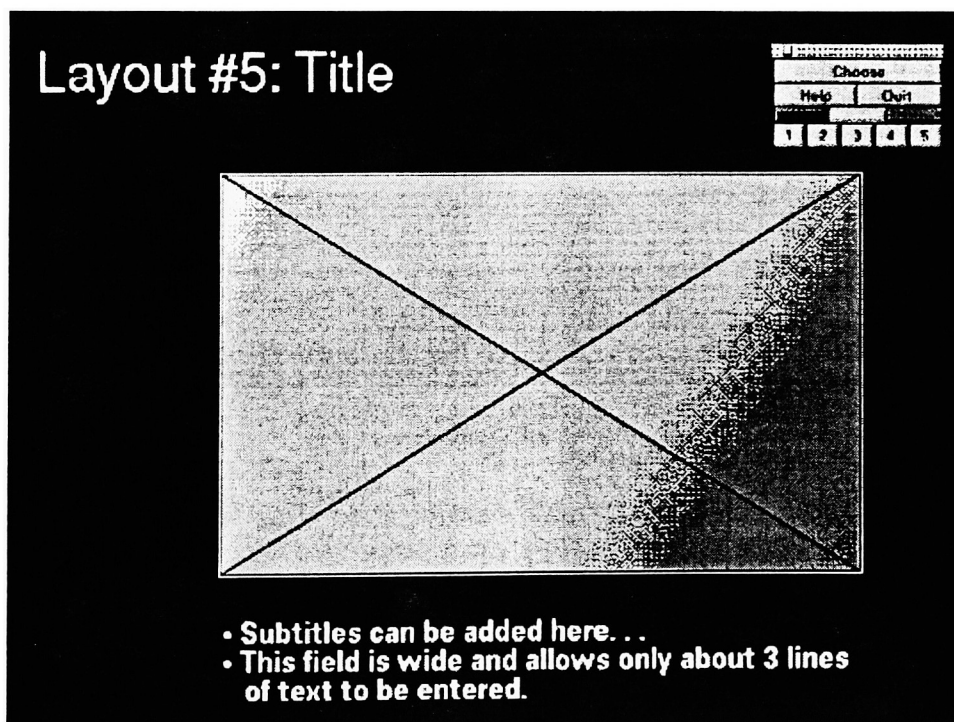
5



- Subtitle goes here: The text field here occupies about half the screen with room left for the addition of a medium-sized image.
- Subtitle goes here: The text field here occupies about half the screen with room left for the addition of a medium-sized image.
- Subtitle goes here: The text field here occupies about half the screen with room left for the addition of a medium-sized image.

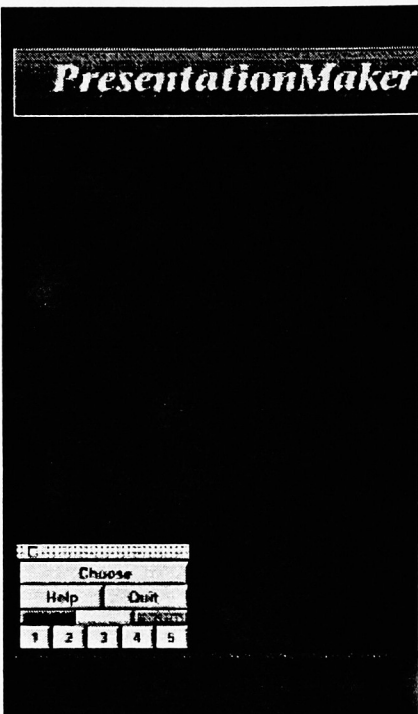


**Layout Choice
#4**



**Layout Choice
#5**

Presentation Maker Help Cards



PresentationMaker

1. **First choose a layout.** The layouts range from #1 (contains the largest text field) to #5, which contains a small text field.

Examine the layouts looking for the one with the relationship between the amount of text and image area that best matches your overall presentation.

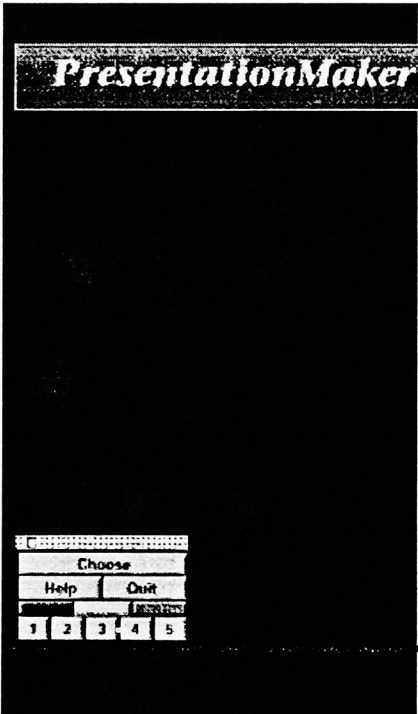
2. **Choose the graphic color.** Click on one of the blue, red, or black and white bars on the palette to compare the alternatives.

3. **Finally, click on "Choose."** After naming your presentation, a new stack will be created for you, based on the choices that you made.

At that time you will be given more control over the more specific details of your layout.

About the Palette
Return

Explanation of Chooser Palette



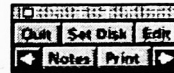
PresentationMaker

- **Choose** creates your presentation using the layout and the colored graphic that you have chosen.
- **Help** brings you to this card.
- **Quit** allows you to exit this program.
- The blue, gray, and red bars change the graphic color on layouts 1, 2, 4 and 5.
- The numbers **1, 2, 3, 4, & 5** allow you to view the five different layouts.

To get the palette back if it disappears for whatever reason, click on the screen.

Getting Started
Return

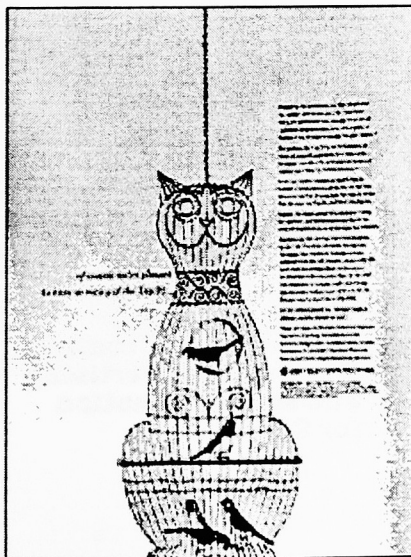
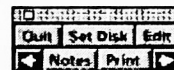
William Golden



- 1911 - 1959
- born in lower Manhattan
- youngest of 12 children

**William Golden
Presentation
created by
Presentation
Maker**

His Education

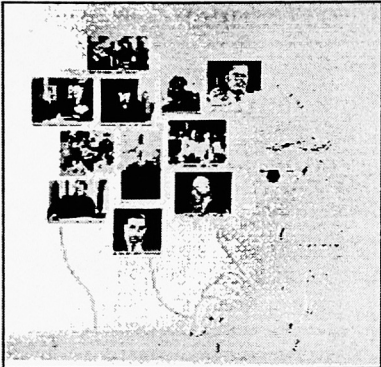


- graduated from Vocational School for Boys at age 17
- took photoengraving and basics of commercial design
- was largely self-educated

**William Golden
Presentation
Example 2**

**William Golden
Presentation
Example 3**


Early Work Experience



- at Los Angeles
Examiner in art dept.
- in New York for
Journal Americas
- for M.E. Agha at Conde
Nast Publications for
House & Garden


**William Golden
Presentation
Example 4**

CBS



- 1937 - joins Columbia
Broadcasting System
- 1940 - named Art
Director at CBS
- 1943 - 1946 - served
as art director
in US Army
- 1951 - named Creative
Director of advertising
and sales promotion
for CBS

New Stature for the Designer




It takes
the right kind
of bail...

The designer
in a corporation
must have a role
not only in the
communication
of ideas
but in the
generation
of ideas
as well.

WEEI

Variety ad for CBS




Whichever
side bail of was
is on the
CBS Television
Network

- double page ad, 1958
- CBS used the adaptable eye symbol widely in its early years to make it memorable.

**William Golden
Presentation
Example 7**


Perspective on Designing



- had a passion for excellence
- responsibility of designer is to communicate client's message
- the best design solution "will look perfectly obvious and inevitable." (p. 33, Pineles)

**William Golden
Presentation
Example 8**

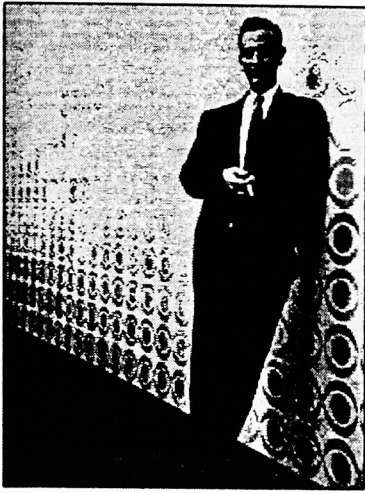
A Passion for Excellence



- Golden set a new aesthetic, excellent standard in American design.
- from announcement folder, 1959.

**William Golden
Presentation
Example 9**

Post-humous Awards



- Golden's death at age 48 was a shock to family and friends.
- named Art Director of the Year by Natl. Soc. of Art Directors in 1959.
- named one of the first members of the NY Art Directors Hall of Fame in 1971.

Quit Set Disk Edit

Notes Print

PresentationMaker

Title	Presenter
Subtitle	Image
Quit	Laserdisc
Invert	Notes
Grid	
Help	Move

Quit Set Disk Edit

Notes Print

Understanding the Director Palette:

Contents:

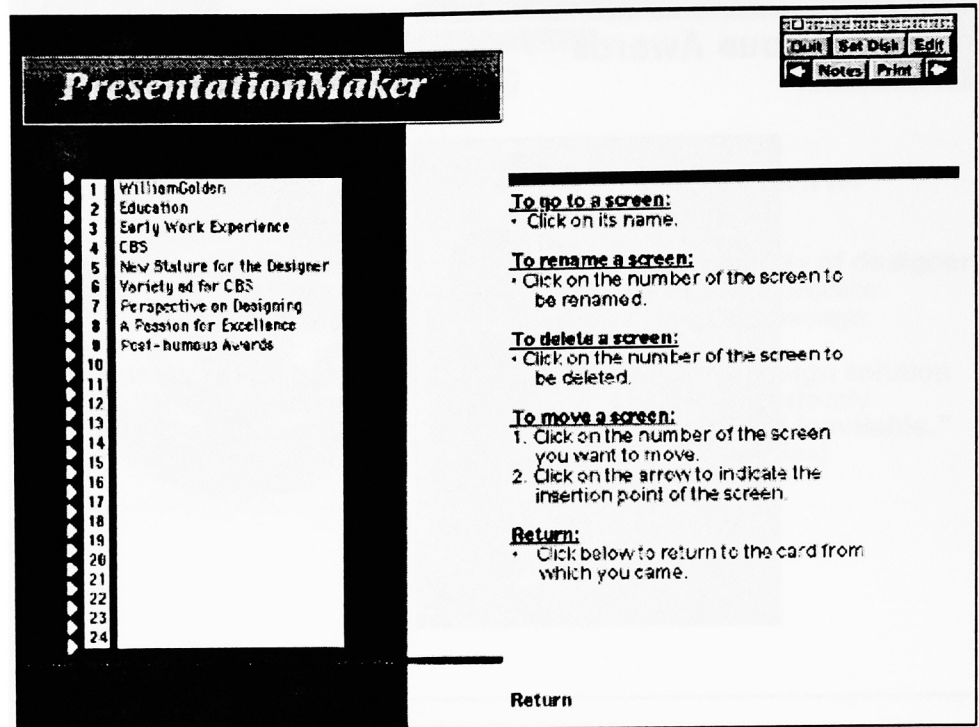
Title Palette	2
Subtitle Palette	2
Presenter Palette	3
Image Palette	3
Quit	4
Laserdisc Images	4
Add	4
Invert	5
Notes	5
Grid	5
Left Arrow	6
Help	6
Move	6
Right Arrow	6

contents 2 3 4 5 6

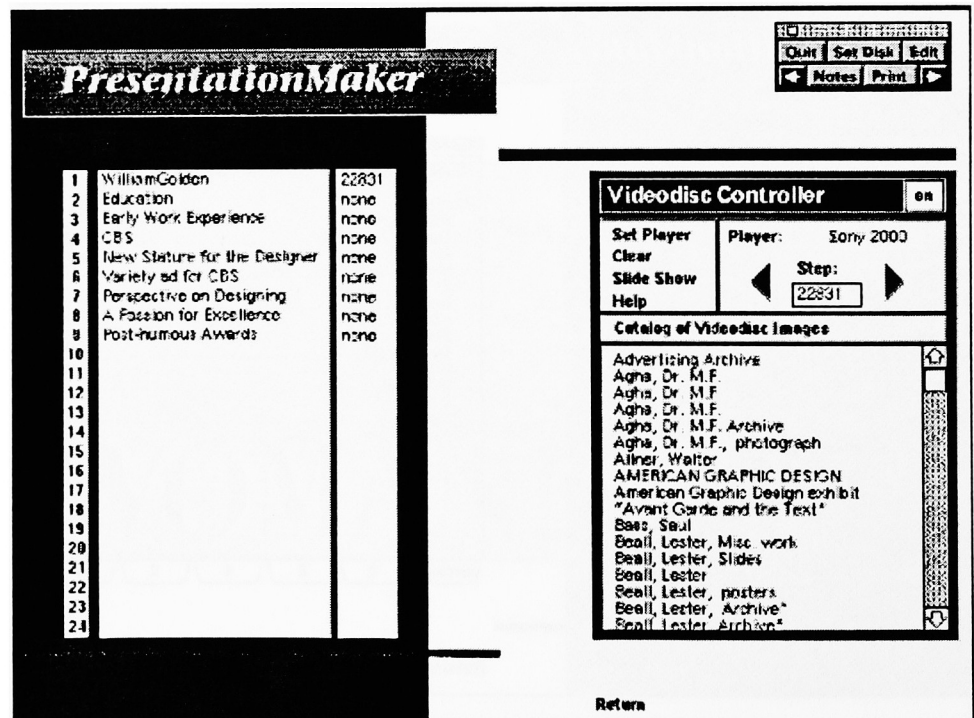
Return

**Help Card
in created
Presentation
Stack**

Move Card in created Presentation Stack



Laserdisc Card in created Presentation Stack



Mehemed Fehmy Agha

Presentation
on Mehemed
Fehmy Agha
created with
Presentation
Maker



- March 3, 1896 - 1978
- born to Turkish parents in the Ukraine
- artistic talents showed at an early age
- sent out of the country at the time of World War I, Agha never saw his parents again

Years at Conde Nast

Presentation
on Mehemed
Fehmy Agha
Example 2

- served as art director
1929 - 1943
- Vanity Fair, Vogue,
& House & Garden
- expected excellence
from his staff



**Presentation
on Mehmed
Fehmy Agha
Example 3**

As Art Director, Agha. . .



- maintained a free creative atmosphere
- encouraged visits to exhibits & museums
- looked for innovations in studio tools & processes

**Presentation
on Mehmed
Fehmy Agha
Example 4**

As Art Director, Agha. . .

- employed the finest photographers
- had expertise himself on photographic technique
- worked closely with his photographers for a particular look



Agha: the designer who could think



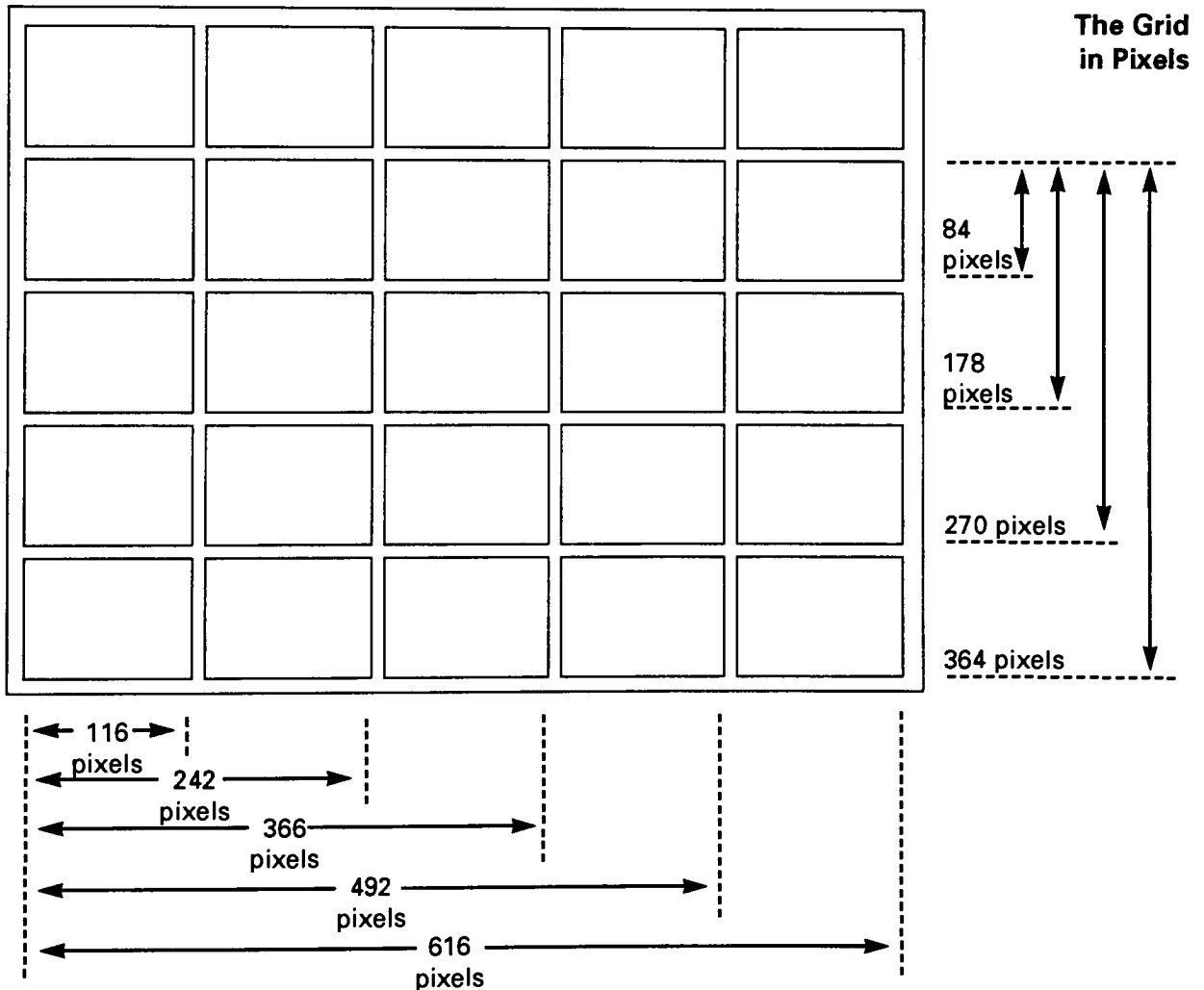
- a man of order, taste and convention
- a strong sense of the practical
- an icy kind of reasoning process
- a finely tuned sense of the absurd

Appendix E: Preparation of PICT Files

To prepare an image file for a PresentationMaker project, adjust the width of the image to match the desired width on the grid area (as pictured below). The height of the image should not exceed 364 pixels, which is the maximum image height. After the image is adjusted, save it as a PICT file using 8 bit color and the system palette.

PICT files that are included in a presentation need to be in the same folder as the presentation itself. If they are not, the presentation will not be able to locate the image and will not run properly.

PICT Files



Appendix F: Thesis Show Materials

Jane Ann Settergren

**Statement
for
Thesis Show**

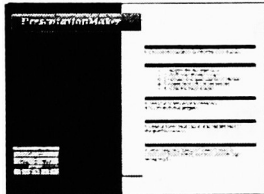
PresentationMaker

Prototype 3.2 in the Graphic Design Archive

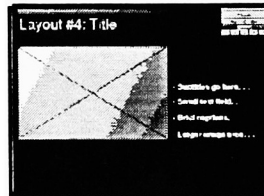
PresentationMaker is an interactive program designed with HyperCard (authoring software that offers simultaneous work with text, image, and sound) that allows a classroom instructor to create or develop a classroom presentation combining images, titles and subtitles, and related notes. The idea was conceived as a module for the Graphic Design Archive project. While the instructor may not be well versed in HyperCard usage, this program provides the necessary tools through the use of palettes that contain choices for the user. After choosing a layout as the basis of the presentation, the user can enter text in the form of titles, subtitles, and/or notes, and also import an image file. The image, as well as the titles and subtitles can be moved according to a predefined grid. The order of the presentation can be rearranged, and screens and notes can be printed. Also each designed screen can be connected with a laserdisc player to show images or information on a second monitor for a two monitor presentation. Presently the laserdisc connection is to the Graphic Design Archive laserdisc.

PresentationMaker is designed to enable the user tremendous capability in the choices of fonts, placement of the elements, and in the use of color and color images, even within the HyperCard environment where color is not well supported. It does this by recording the choices made by the user and completing programming for the user "behind the scenes," making the presentation relatively easy to use as well as effective.

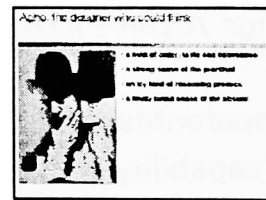
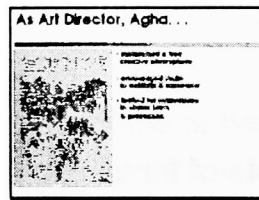
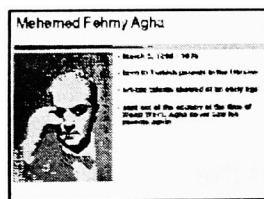
PresentationMaker



1. After making a layout choice, a new presentation is generated. It will provide the tools appropriate for entering and designing the content of the presentation.



2. The user can enter and manipulate text and images to create their own customized presentation.



**Wall Exhibit
for
Thesis Show**

Appendix G: Scripts

SCRIPTS FOR STACK: PresentationMaker

=====

```
on openstack
  global ChooserGlobal,pictState
  put "false" into pictState
  set cursor to watch
  hide menubar
  —set the userLevel to 2
  if not (chooserGlobal contains "true") then
    palette "Chooser"
    set loc of window "Chooser" to 8,365
  end if
end openStack

on closeStack
  global ChooserGlobal
  if chooserGlobal contains "true" then
    send "close" to window "chooser"
  end if
end closeStack

on openCard
  hide menubar
end openCard

on closeCard
  global pictState,myWindow
  if pictState contains "true" then
    close window myWindow
  end if
  put "false" into pictState
end closeCard

on mouseUp
  checkPalette
end mouseUp

on CheckPalette
  —this handler is written as a way to recover
  —after the user closes the palette that is needed
  —also to move the palette if it is hidden
  global ChooserGlobal
  if not (ChooserGlobal contains "true") then
    palette "Chooser"
  end if
end CheckPalette

on layout1
  global CdName,pictName,pictLoc,layoutNum
  global pictState,myWindow>windowLoc
  hide menubar
  CheckPalette
  set loc of window "Chooser" to 511,20
```

**Stack Script
for
Presentation
Maker**

**CheckPalette
recovers
palette if it is
hidden**

**Handler for
first Layout
Card**

```
—Programming for colored graphic element
put 1 into layoutNum
put the short name of this cd into CdName
```

```
put "1.Blue" into pictName
put "13,89" into pictLoc
```

```
—Programming for picture file
put "true" into pictState
put "Image1" into myWindow
put "12,105" into windowLoc
picture myWindow,file,rect,false
put item 1 of windowLoc into xLoc
put item 2 of windowLoc into yLoc
set loc of window myWindow to xLoc,yLoc
set visible of window myWindow to true
end layout1
```

Handler for Second Layout Card

```
on layout2
global CdName,pictName,pictLoc,layoutNum
global pictState,myWindow,windowLoc
```

```
hide menubar
CheckPalette
set loc of window "Chooser" to 511,20
```

```
—Programming for colored graphic element
put 2 into layoutNum
put the short name of this cd into CdName
```

```
put "2.red" into pictName
put "12,85" into pictLoc
```

```
—Programming for picture file
put "true" into pictState
put "Image2" into myWindow
put "12,105" into windowLoc
picture myWindow,file,rect,false
put item 1 of windowLoc into xLoc
put item 2 of windowLoc into yLoc
set loc of window myWindow to xLoc,yLoc
set visible of window myWindow to true
end layout2
```

Handler for Third Layout Card

```
on layout3
global CdName,pictName,pictLoc,layoutNum
global pictState,myWindow,windowLoc
```

```
hide menubar
CheckPalette
set loc of window "Chooser" to 511,20
```

```
—Programming for colored graphic element
put 3 into layoutNum
put the short name of this cd into CdName
```

```
put empty into pictName
put empty into pictLoc
```

```

—Programming for picture file
put "true" into pictState
put "Image2" into myWindow
put "12,105" into windowLoc
picture myWindow,file,rect,false
put item 1 of windowLoc into xLoc
put item 2 of windowLoc into yLoc
set loc of window myWindow to xLoc,yLoc
set visible of window myWindow to true
end layout3

```

```

on layout4
global CdName,pictName,pictLoc,layoutNum
global pictState,myWindow,windowLoc

```

```

hide menubar
CheckPalette
set loc of window "Chooser" to 511,20

```

```

—Programming for colored graphic element
put 4 into layoutNum
put the short name of this cd into CdName

```

```

put "4.red" into pictName
put "13,13" into pictLoc

```

```

—Programming for picture file
put "true" into pictState
put "Image4" into myWindow
put "12,105" into windowLoc
picture myWindow,file,rect,false
put item 1 of windowLoc into xLoc
put item 2 of windowLoc into yLoc
set loc of window myWindow to xLoc,yLoc
set visible of window myWindow to true
end layout4

```

```

on layout5
global CdName,pictName,pictLoc,layoutNum
global pictState,myWindow,windowLoc

```

```

hide menubar
CheckPalette
set loc of window "Chooser" to 511,30

```

```

—Programming for colored graphic element
put 5 into layoutNum
put the short name of this cd into CdName

```

```

put "5.Blue" into pictName
put "12,12" into pictLoc

```

```

—Programming for picture file
put "true" into pictState
put "image5" into myWindow
put "137,105" into windowLoc
picture myWindow,file,rect,false
put item 1 of windowLoc into xLoc

```

Handler for Fourth Layout Card

Handler for Fifth Layout Card

```

put item 2 of windowLoc into yLoc
set loc of window myWindow to xLoc,yLoc
set visible of window myWindow to true
end layout5

```

**"On Idle"
Handler
displays
picture
resource files**

```

on idle
  global pictName, pictLoc,cdName

  if the short name of this bg is "pMaker" then
    showPict "Titlebar.la", 6,25
    showPict "Titlestrip", 6,427
  end if
  if not (pictName is empty) then
    if short name of this cd = cdName then
      put item 1 of pictLoc into xLoc
      put item 2 of pictloc into yLoc
      showPict pictName,xLoc,yLoc
    end if
  end if
end idle

```

**Scripts for
Chooser
Palette**

```

—*****
—  Chooser Palette
—*****

on temp1
  —*****
  —view Layout #1
  —*****
  play "click"
  go cd "Layout1"
end temp1

on temp2
  —*****
  —view Layout #2
  —*****
  play "click"
  go cd "Layout2"
end temp2

on temp3
  —*****
  —view Layout #3
  —*****
  play "click"
  go cd "Layout3"
end temp3

on temp4
  —*****
  —view Layout #4
  —*****
  play "click"
  go cd "Layout4"
end temp4

```

```

on temp5
  —*****
  —view Layout #5
  —*****
  play "click"
  go cd "Layout5"
end temp5

on helpCd
  —*****
  —go to help card
  —*****
  play "click"
  push card
  lock screen
  go cd "Help1"
  unlock screen with visual wipe right
end helpCd

on chooseCd
  —*****
  —creates new stack using chosen layout
  —*****
  global pictLoc, pictName,cdName,temp
  global pictState,myWindow,layoutNum

  play "click"

  —check to see if the card is a template
  if not (cdName contains "Layout") then
    beep
    Answer "Click on a number to see the layouts. " with "OK"
    checkPalette
    exit chooseCd
  end if

  —verify choice
  Answer "Create presentation with this layout?" with~
  "Continue" or "Cancel"
  if it = "Cancel" then
    exit chooseCd
  end if

  —get name for newStack
  Ask "What is the name of your presentation?" with "PresentationName"
  put it into newStack
  put the short name of this stack into thisStack
  hide button "hiliter"

  —Close picture window
  close window myWindow
  put "false" into pictState

  —Set up global variables for script of newStack
  put " put " & quote & pictName &quote&" into pictName"~
  into line 4 of cd fld "newScript" of cd "help" —PictName
  put " put " & quote& pictLoc &quote&" into pictLoc"~
  into line 5 of cd fld "newScript" of cd "help" —PictLoc

```

**ChooseCd
Creates the
New
Presentation**

ChooseCd cont.

```
put empty into newScript

— set up sequence for animated cursor
put “00,07,15,22,30,37,45,52” into seq
put 8 into seqNum

—get actual script for stack
repeat with x = 1 to the number of lines of cd fld “newScript” of cd “help”
  put line x of cd fld “newScript” of cd “help” after newScript
  put return after newScript
  put (x div 5) * 5 into test
  if x = test then
    add 1 to seqNum
    if seqNum >8 then put 1 into seqNum
    set cursor to “B9:” & item seqNum of seq
  end if
end repeat

—create the newStack
lock screen
set cursor to busy
create stack newStack with bg “layoutBG” & layoutNum

set the script of stack newStack to newScript

—prepare to copy layout cd for new stack
go cd “layout”&layoutNum of stack thisStack

—get card script for new stack
put the script of cd “layout”&layoutNum into temp
put empty into newScript
repeat with x = 1 to the number of lines of cd fld “cardScript” of cd “helpGuide”
  put line x of cd fld “cardScript” of cd “helpGuide” after newScript
  put return after newScript
  put (x div 5) * 5 into test
  if x = test then
    add 1 to seqNum
    if seqNum >8 then put 1 into seqNum
    set cursor to “B9:” & item seqNum of seq
  end if
end repeat
set script of cd “Layout” & layoutNum to newScript

—copy layout card and paste into new stack
go cd “Layout”& layoutNum
doMenu “Copy Card”
go stack newStack
doMenu “Paste Card”
go cd 1 of stack newStack
doMenu “Delete Card”

—set up of newly created layout card
set cantDelete of this cd to false
select cd btn “hiliterSwitch”
```

**ChooseCd
cont.**

```
doMenu "Clear Button"
select cd btn "grid"
doMenu "Clear Button"
put newstack into line 1 of cd fld 1
set the name of this cd to newStack

—reset the layout card
go cd "layout"&layoutNum of stack thisStack
set script of cd "Layout"&layoutNum to temp
show button "hiliter" of cd "Layout"&layoutNum

—copy help card and paste into new stack
go cd "HelpGuide"
doMenu "Copy Card"
go stack newStack
doMenu "Paste Card"

—copy move card and paste into new stack
go cd "MoveCard" of stack thisStack
doMenu "Copy Card"
go stack newStack
go last
doMenu "Paste Card"

—copy Laserdisc Images card and paste into new stack
go cd "Laserdisc Images" of stack thisStack
doMenu "Copy Card"
go stack newStack
go last
doMenu "Paste Card"

play "crystal"
go first cd of stack newStack
choose browse tool
end chooseCd
```

```
on blueColor
—*****
—displays blue graphic for that card
—*****

global pictName, pictLoc, CdName
play "click"
if cardName contains "3" then
    put empty into pictName
    put empty into pictLoc
else
    if CdName contains "1" then
        put "1.Blue" into pictName
        put "13,89" into pictLoc
    else
        if cdName contains "2" then
            put "2.Blue" into pictName
            put "12,85" into pictLoc
        else
            if cdName contains "4" then
                put "4.Blue" into pictName
```

**blueColor
sets the
appropriate
graphic**

```

        put "13,13" into pictLoc
    else
        if cdName contains "5" then
            put "5.Blue" into pictName
            put "12,12" into pictLoc
        end if
    end if
end if
end if
end if
end blueColor

```

**redColor
sets the
appropriate
graphic
to red**

```

on redColor
    _____
    —displays red graphic for that card
    _____
    global pictName, pictLoc, cdName
    play "click"
    if cardName contains "3" then
        put empty into pictName
        put empty into pictLoc
    else
        if cdName contains "1" then
            put "1.Red" into pictName
            put "13,89" into pictLoc
        else
            if cdName contains "2" then
                put "2.Red" into pictName
                put "12,85" into pictLoc
            else
                if cdName contains "4" then
                    put "4.Red" into pictName
                    put "13,13" into pictLoc
                else
                    if cdName contains "5" then
                        put "5.Red" into pictName
                        put "12,12" into pictLoc
                    end if
                end if
            end if
        end if
    end if
end if
end redColor

```

**grayColor
selects no
graphic**

```

on grayColor
    _____
    —allows bitmap graphic to show
    _____
    global pictName, pictLoc
    play "click"
    put empty into pictName
    put empty into pictLoc
    lock screen
    suspend idle
    unlock screen
end grayColor

```

```

on quitMe
  —*****
  —quits HyperCard
  —*****
  global pictState,myWindow
  play "click"
  answer "Quit Program?" with "Yes" or "No" or "Cancel"
  if it = "cancel" then
    exit quitMe
    if it = "Yes" then
      put "false" into pictState
      doMenu "quit HyperCard"
    else
      if it = "no" then
        go first
      end if
    end if
  end if
end quitMe

```

quitMe
allows user
to exit
HyperCard

**** CARD #1: TitleCd *******

```

on openCard
  playMovie "Intro1.mm"
end openCard

```

Card Scripts
from
Presentation
Maker

**** CARD #2: Layout1 *******

```

on openCard
  global cdName
  put the short name of this cd into cdName
  layout1
end openCard

```

**** CARD #3: Layout2 *******

```

on openCard
  global cdName
  put the short name of this cd into cdName
  layout2
end openCard

```

**** CARD #4: Layout3 *******

```

on openCard
  global cdName
  put the short name of this cd into cdName
  layout3
end openCard

```

**** CARD #5: Layout4 *******

```

on openCard
  global cdName
  put the short name of this cd into cdName
  layout4
end openCard

```

**** CARD #6: Layout5 *******

```

on openCard
  global cdName
  put the short name of this cd into cdName
  layout5
end openCard

```

Card Button Scripts for Layout Cards

```
** CARDS #2- 6, BUTTON #1 *****
on mouseUp
  global showGrid
  if showGrid = "true" then
    hideGrid
  else
    showGrid
  end if

end mouseUp

on hideGrid
  global showGrid
  put "false" into showGrid
  repeat with x = 1 to 25
    set the style of bg btn x to opaque
  end repeat
end hideGrid

on showGrid
  global showGrid
  put "true" into showGrid
  repeat with x = 1 to 25
    set the style of bg btn x to transparent
  end repeat
end showGrid

** CARDS # 2- 6, BUTTON #2: HiliterSwitch
*****

on mouseUp
  show cd btn "hiliter"
end mouseUp

** CARDS # 2- 6, BUTTON #3: hiliter
*****

on mouseUp
  set the hilite of me to true
  hide me
end mouseUp
```

Cards 7,8,9 are copied into new presentation

```
** CARDS #7 ,8,& 9 are pasted into the new presentation and Scripts are
found in the William Golden Presentation Scripts.
*****
```

```
** CARD #10: Help1 *****
on openCard
  set loc of window "Chooser" to 8,365
end openCard
```

HelpCard #1 Scripts

```
** CARD ** 10, BUTTON #1: About the Palette
*****
on mouseUp
  lock screen
  go cd "aboutPalette"
  unlock screen with visual wipe right
end mouseUp
```

```
** CARD ** 10, BUTTON #2: Return
*****
on mouseUp
  lock screen
  pop cd
  unlock screen with visual wipe left
end mouseUp
```

```
** CARD #11: AboutPalette *****
on openCard
  set loc of window "Chooser" to 8,365
end openCard
```

HelpCard #2 Scripts

```
** CARD ** 11, BUTTON #1: Getting Started
*****
on mouseUp
  lock screen
  go cd "help1"
  unlock screen with visual wipe left
end mouseUp
```

```
** CARD ** 11, BUTTON #2: Return
*****
on mouseUp
  lock screen
  pop cd
  unlock screen with visual wipe left
end mouseUp
```

** STACK SCRIPT *****

```
on openStack
  global pictName,pictLoc
  global showGrid,counting,myMode
  put "4.Blue" into pictName
  put "13,13" into pictLoc
  put "false" into showGrid

  —set the userLevel to 2
  get the number of cds in this stack
  put it into counting
  subtract 2 from counting
  hide menubar
  get line 1 of bg fld "mode" of cd 1
  put it into myMode
  if myMode contains "edit" then
    palette "Director"
    set loc of window "Director" to 512,16
  else
    palette "Presenter"
    set loc of window "Presenter" to 512,16
  end if
end openStack
```

```
on closeStack
  global directorGlobal,presenterGlobal
  if directorGlobal contains "true" then
    send "close" to window "director"
  end if
  if presenterGlobal contains "true" then
    send "close" to window "presenter"
  end if
  paletteCloser
end closeStack
```

```
on paletteCloser
  global TitleGlobal
  global subtitleGlobal, ImageGlobal
  if titleGlobal contains "true" then
    send "close" to window "title"
  end if
  if subtitleGlobal contains "true" then
    send "close" to window "subtitle"
  end if
  if imageGlobal contains "true" then
    send "close" to window "image"
  end if
end paletteCloser
```

```
on mouseUp
  checkPalette
end mouseUp
```

```
on closeCard
  global pictState,myWindow,myNotes
  if myNotes = "on" then
    hide cd fld "textEntry"
    set visible of window myWindow to true
    put "off" into myNotes
  end if
  if pictState = "true" then
    close window myWindow
    put "false" into pictState
  end if
end closeCard
```

```
on CheckPalette
  global myMode
  if myMode = "edit" then
    palette "director"
  else
    palette "presenter"
  end if
end CheckPalette
```

```
on idle
  global pictName, pictLoc
  if the short name of this bg is "pMaker" then
    showPict "Titlebar.la", 6,25
    showPict "Titlestrip". 6,427
  else
    if not (pictName is empty) then
      put item 1 of pictLoc into Xloc
      put item 2 of pictloc into Yloc
      showPict pictName,xLoc,yLoc
    end if
  end if
end idle
```

```
—*****
— Director Palette
—*****
```

**Scripts for
Director
Palette**

```
on invertScreen
  global bgColor
  play "click"
  put the short name of this bg into current
  if current = "pMaker" then
    beep
    answer "Sorry, Invert does not work here."
    exit invertScreen
  end if
  if bgColor = "black" then
    hide cd btn "hiliter" of this cd
    put "white" into bgColor
  else
    show cd btn "hiliter" of this cd
    set the hilite of cd btn "hiliter" of this cd to true
    put "black" into bgColor
  end if
end invertScreen
```

**invertScreen
reverses the
background
color**

**gridTog
shows/hides
the grid**

```
on gridTog
  global showGrid
  play "click"
  put the short name of this bg into current
  if current = "pMaker" then
    beep
    answer "Sorry, the grid does not work here."
    exit gridTog
  end if
  if showGrid = "true" then
    put "false" into showGrid
    repeat with x = 1 to 25
      set the style of bg btn x to opaque
    end repeat
  else
    put "true" into showGrid
    repeat with x = 1 to 25
      set the style of bg btn x to transparent
    end repeat
  end if
end gridTog

on helpGuide
  play "click"
  push this card
  paletteCloser
  go cd "HelpGuide"
end helpGuide

on quitMe
  global pictState
  play "click"
  answer "Quit HyperCard?" with "Yes" or "No" or "Cancel"
  if it = "Yes" then
    put "false" into pictState
    doMenu "quit HyperCard"
  else
    if it = "no" then
      go cd 1 of stack "PresentationMaker"
    end if
  end if
end quitMe
```

**TitleGuide
displays Title
Palette**

```
on TitleGuide
  global titleGlobal
  play "click"
  put the short name of this bg into current
  if current = "pMaker" then
    beep
    answer "Sorry, this palette does not work here."
    exit titleGuide
  end if
  if not (titleGlobal contains "true") then
    Palette "Title", "512,124"
  else
    send "close" to window "title"
  end if
end TitleGuide
```

```

on BodyGuide
  global subtitleGlobal
  play "click"
  put the short name of this bg into current
  if current = "pMaker" then
    beep
    answer "Sorry, this palette does not work here."
    exit BodyGuide
  end if
  if not (subtitleGlobal contains "true") then
    palette "Subtitle", "512,233"
  else
    send "close" to window "subtitle"
  end if
end BodyGuide

```

BodyGuide
displays
Subtitle
Palette

```

on importImages
  global imageGlobal
  play "click"
  put the short name of this bg into current
  if current = "pMaker" then
    beep
    answer "Sorry, this palette does not work here."
    exit importImages
  end if
  if not (imageGlobal contains "true") then
    palette "Image", "512,343"
  else
    send "close" to window "image"
  end if
end importImages

```

importImages
displays Image
Palette

```

on viewScreen
  global directorGlobal
  global presenterGlobal, myMode
  play "click"
  if directorGlobal contains "true" then
    send "close" to window "director"
  end if
  paletteCloser
  palette "presenter", "512,16"
  put "presenter" into bg fld "mode" of cd 1
  put "presenter" into myMode
end viewScreen

```

ViewScreen
displays
Presenter
Palette

```

on myNotes
  global myNotes, pictState, myWindow
  play "click"
  put the short name of this bg into current
  if current = "pMaker" then
    beep
    answer "Sorry, there are NO notes here."
    exit myNotes
  end if
  if myNotes contains "off" then
    paletteCloser
    if pictState contains "true" then
      set visible of window myWindow to false
    end if
  end if
end myNotes

```

myNotes
opens field
for related
notes

myNotes
cont.

```
end if
show cd fld "textEntry"
put "on" into myNotes
play "crystal"
else
hide cd fld "textEntry"
if pictState contains "true" then
set visible of window myWindow to true
end if
put "off" into myNotes
play "crystal"
end if
end myNotes
```

```
on addLaserdisc
play "click"
push card
paletteCloser
go cd "Laserdisc Images"
end addLaserdisc
```

addScreen
adds another
screen to
presentation

```
on addScreen
global counting,cdName
play "click"
if counting = 25 then
answer "Sorry, 24 screens is the limit." with "OK"
exit addScreen
end if
```

```
put the short name of this bg into current
if current = "pMaker" then
beep
answer "Sorry, ADD does not work here."
exit addScreen
end if
```

```
ask "Please enter a name for the next screen." ~
with "NewScreen"&Counting
if it is empty then
exit addScreen
else
put it into ScreenName
end if
```

```
doMenu "Copy Card"
doMenu "Paste Card"
```

```
add 1 to counting
set the name of this cd to ScreenName
put the short name of this cd into cdName
```

```
delete line 1 of cd fld 1
put screenName into line 1 of cd fld 1
play "crystal"
end addScreen
```

```

on doMenu whatever
  if whatever contains "new card" then
    addScreen
  else
    if whatever contains "quit" then
      go cd 1 of stack "PresentationMaker"
    else
      pass doMenu
    end if
  end if
end doMenu

```

**doMenu
filters menu
choices**

```

on moveScreen
  play "click"
  push this card
  go cd "moveCard"
end moveScreen

```

```

on prevScreen
  play "click"
  go prev
end prevScreen

```

```

on nextScreen
  play "click"
  go next
end nextScreen

```

```

—*****
— Title Field Palette
—*****

```

**Scripts
for
Title Field
Palette**

```

on HelveticaT
  play "click"
  set the textFont of cd fld "Title" to "helvetica"
  repeat with counter = 1 to the number of words in cd fld 1
    set the textfont of word counter of cd fld 1 to Helvetica
  end repeat
end HelveticaT

```

```

on AvantGardeT
  play "click"
  set the textFont of cd fld "Title" to "Avant Garde"
  repeat with counter = 1 to the number of words in cd fld 1
    set the textfont of word counter of cd fld 1 to Avant Garde
  end repeat
end AvantGardeT

```

```

on TimesT
  play "click"
  set the textFont of cd fld "Title" to "Times"
  repeat with counter = 1 to the number of words in cd fld 1
    set the textfont of word counter of cd fld 1 to Times
  end repeat
end TimesT

```

**Scripts for
Title Field
Palette
cont.**

```
on BookmanT
  play "click"
  set the textFont of cd fld "Title" to "Bookman"
  repeat with counter = 1 to the number of words in cd fld 1
    set the textfont of word counter of cd fld 1 to Bookman
  end repeat
end BookmanT
```

```
on PlainT
  play "click"
  set the textStyle of cd fld "Title" to "Plain"
  repeat with counter = 1 to the number of words in cd fld 1
    set the textstyle of word counter of cd fld 1 to plain
  end repeat
end PlainT
```

```
on BoldT
  play "click"
  set the textStyle of cd fld "Title" to "Bold"
  repeat with counter = 1 to the number of words in cd fld 1
    set the textstyle of word counter of cd fld 1 to bold
  end repeat
end BoldT
```

```
on Title24
  play "click"
  set the textSize of cd fld "Title" to 24
  repeat with counter = 1 to the number of words in cd fld 1
    set the textsize of word counter of cd fld 1 to 24
  end repeat
end Title24
```

```
on Title36
  play "click"
  set the textSize of cd fld "Title" to 36
  repeat with counter = 1 to the number of words in cd fld 1
    set the textsize of word counter of cd fld 1 to 36
  end repeat
end Title36
```

**Scripts
for moving
the Fields**

```
on upT
  global moveMe
  play "click"
  put "cd fld title" into moveMe
  moveUp
end upT
```

```
on moveUp
  global moveMe, pictName
  put the loc of moveMe into myLoc
  put the height of moveMe div 2 into myHeight
  put char 1 of pictName into myNum
```

```
—check for location
put item 2 of myLoc into ycord
if ycord - myHeight < 96 then
  beep
  answer "This field can move no farther." with "OK"
```

```
    exit moveUp
end if

subtract 93 from ycord

if ycord - myHeight < 105 then
    if myNum = 4 then
        set the loc of moveMe to item 1 of the loc-
        of moveMe, 32+myHeight
    else
        set the loc of moveMe to item 1 of the loc-
        of moveMe, 12 +myHeight
    end if
else
    if ycord - myHeight < 198 then
        set the loc of moveMe to item 1 of the loc-
        of moveMe, 105 +myHeight
    else
        if ycord - myHeight < 291 then
            set the loc of moveMe to item 1 of the loc-
            of moveMe, 198 +myHeight
        else
            if ycord - myHeight < 384 then
                set the loc of moveMe to item 1 of the-
                loc of moveMe, 291 +myHeight
            else
                set the loc of moveMe to item 1-
                of the loc of moveMe, 384 +myHeight
            end if
        end if
    end if
end if
end moveUp

on downT
    global moveMe
    play "click"
    put "cd fld title" into moveMe
    moveDown
end downT

on moveDown
    global moveMe
    put the loc of moveMe into myLoc
    put the height of moveMe div 2 into myHeight
    put item 2 of myLoc into yCord
    if yCord + myHeight > 384 then
        beep
        answer "This field can move no farther." with-
        "OK"
        exit moveDown
    end if
    add 93 to ycord

    if ycord - myHeight < 198 then
        set the loc of moveMe to item 1 of the loc-
        of moveMe, 105 + myHeight
    else
```

**Scripts
for moving
the Fields
cont.**

```
if ycord - myHeight < 291 then
  set the loc of moveMe to item 1 of the loc-
  of moveMe, 198 + myHeight
else
  if ycord - myHeight < 384 then
    set the loc of moveMe to item 1 of the loc-
    of moveMe, 291 + myHeight
  else
    set the loc of moveMe to item 1 of the loc-
    of moveMe, 384 + myHeight
  end if
end if
end if
end moveDown

on rtT
  global moveMe
  play "click"
  put "cd fld title" into moveMe
  moveRight
end rtT

on moveRight
  global moveMe
  put the loc of moveMe into myLoc
  put the width of moveMe div 2 into myWidth

  put item 1 of myLoc into xCord
  if xCord + myWidth > 512 then
    beep
    answer "this field can move no farther" with "OK"
    exit moveRight
  end if
  add 125 to xcord

  if xcord - myWidth < 262 then
    set the loc of moveMe to 137 + myWidth, item 2 of the loc-
    of moveMe
  else
    if xcord - myWidth < 387 then
      set the loc of moveMe to 262 + myWidth, item 2 of the loc-
      of moveMe
    else
      set the loc of moveMe to 387 + myWidth, item 2 of the loc-
      of moveMe
    end if
  end if
end moveRight

on ltT
  global moveMe
  play "click"
  put "cd fld title" into moveMe
  moveLeft
end ltT

on moveLeft
  global moveMe
```

**Scripts
for moving
the Fields
cont.**

```

put the loc of moveMe into myLoc
put the width of moveMe div 2 into myWidth
put item 1 of myLoc into xCord
if xCord - myWidth < 136 then
  beep
  answer "this field can move no farther" with "OK"
  exit moveLeft
end if
subtract 125 from xcord

if xcord - myWidth < 137 then
  set the loc of moveMe to 12 + myWidth, item 2 of the loc-
  of moveMe
else
  if xcord - myWidth < 262 then
    set the loc of moveMe to 137 + myWidth, item 2 of the loc-
    of moveMe
  else
    set the loc of moveMe to 262 + myWidth, item 2 of the loc-
    of moveMe
  end if
end if
end moveLeft

```

```

—*****
— Body Text Field Palette
—*****

```

**Scripts
for Subtitle
Palette**

```

on HelveticaB
  play "click"
  set the textFont of cd fld "bodyText" to "helvetica"
  repeat with counter = 1 to the number of lines in cd fld 2
    set the textFont of line counter of cd fld 2 to helvetica
  end repeat
end HelveticaB

on AvantGardeB
  play "click"
  set the textFont of cd fld "bodyText" to "Avant Garde"
  repeat with counter = 1 to the number of lines in cd fld 2
    set the textFont of line counter of cd fld 2 to Avant Garde
  end repeat
end AvantGardeB

on TimesB
  play "click"
  set the textFont of cd fld "bodyText" to "Times"
  repeat with counter = 1 to the number of lines in cd fld 2
    set the textFont of line counter of cd fld 2 to Times
  end repeat
end TimesB

on BookmanB
  play "click"
  set the textFont of cd fld "bodyText" to "Bookman"
  repeat with counter = 1 to the number of lines in cd fld 2
    set the textFont of line counter of cd fld 2 to Bookman
  end repeat
end repeat

```

**Scripts
for Subtitle
Palette
cont.**

```
end BookmanB

on PlainB
  play "click"
  set the textStyle of cd fld "bodyText" to "Plain"
  repeat with counter = 1 to the number of lines in cd fld 2
    set the textStyle of line counter of cd fld 2 to plain
  end repeat
end PlainB

on BoldB
  play "click"
  set the textStyle of cd fld "bodyText" to "Bold"
  repeat with counter = 1 to the number of lines in cd fld 2
    set the textStyle of line counter of cd fld 2 to bold
  end repeat
end BoldB

on body18
  play "click"
  set the textSize of cd fld "bodyText" to 18
  repeat with counter = 1 to the number of lines in cd fld 2
    set the textSize of line counter of cd fld 2 to 18
  end repeat
end body18

on body24
  play "click"
  set the textSize of cd fld "bodyText" to 24
  repeat with counter = 1 to the number of lines in cd fld 2
    set the textSize of line counter of cd fld 2 to 24
  end repeat
end body24

on upB
  global moveMe
  play "click"
  put "cd fld bodyText" into moveMe
  moveUp
end upB

on downB
  global moveMe
  play "click"
  put "cd fld bodyText" into moveMe
  moveDown
end downB

on rtB
  global moveMe
  play "click"
  put "cd fld bodyText" into moveMe
  moveRight
end rtB

on ltB
  global moveMe
  play "click"
```

```
put "cd fld bodyText" into moveMe
moveLeft
end ItB
```

```
—*****
```

```
— Import Window Palette
```

```
—*****
```

```
on importGraphic
  global cdName,pictState,myWindow>windowLoc
  play "click"
  put the short name of this cd into cdName
  put "12,105" into windowLoc
```

```
if pictState contains "true" then
  answer "Replace the current picture?"~
  with "yes" or "no image" or "cancel"
  if it = "cancel" then
    exit importGraphic
  end if
  close window myWindow
  put false into pictState
end if
```

```
—rewrite script for NO IMAGE
```

```
if it = "no image" then
  —Programming for picture file into card script
  put " put "&quote& "false" &quote& " into pictState"~
  into line 11 of cd fld "cardScript" of cd "helpGuide"
  newCdScript
  exit importGraphic
end if
```

```
—get the file name of the picture
put "Display which?" into prompt
answer file prompt of type PICT
if it is empty then exit to hyperCard
put the number of characters of it into totalchar
put empty into tester
repeat until tester = ":"
  subtract 1 from totalchar
  put char totalchar of it into tester
end repeat
repeat with counter = 1 to totalchar
  delete char 1 of it
end repeat
put it into myWindow
```

```
—Programming for picture file into card script
put " put "&quote& "true" &quote& " into pictState"~
into line 11 of cd fld "cardScript" of cd "helpGuide"
put " put "&quote& myWindow &quote& " into myWindow"~
into line 13 of cd fld "cardScript" of cd "helpGuide"
put " put " &quote & windowLoc &quote& " into windowLoc"~
into line 14 of cd fld "cardScript" of cd "helpGuide"
newCdScript
—show picture window
put "true" into pictState
picture myWindow,file,rect,false
```

```

put item 1 of windowLoc into xLoc
put item 2 of windowLoc into yLoc
set loc of window myWindow to xLoc,yLoc
set visible of window myWindow to true
end importGraphic

```

Scripts for moving Image Window

```

on gphUp
  global pictstate,myWindow
  global moveMe,myNum
  play "click"
  if pictState = "true" then
    put "window "&myWindow &" into moveMe
  else
    answer "No window open on this card." with "OK"
    exit gphUp
  end if
  put the loc of moveMe into myLoc
  put item 1 of myLoc into xCord
  put item 2 of myLoc into yCord
  if yCord < 96 then
    beep
    answer "This image can move no farther." with "OK"
    exit gphUp
  end if
  subtract 93 from yCord

  if yCord < 105 then
    if myNum = 4 then
      set the loc of moveMe to xCord, 32+yCord
    else
      set the loc of moveMe to xCord, 12
    end if
  else
    if yCord < 198 then
      set the loc of moveMe to xCord,105
    else
      if yCord < 291 then
        set the loc of moveMe to xCord, 198
      else
        set the loc of moveMe to xCord, 291
      end if
    end if
  end if
  —reset location of window on card Script
  put " put "&myWindow &" into myWindow"~
  into line 13 of cd fld "cardScript" of cd "helpGuide"
  put xCord& ","& yCord into windowLoc
  put " put " &"& windowLoc &" into windowLoc" into~
  line 14 of cd fld "cardScript" of cd "helpGuide"
  newCdScript
end gphUp

on gphD
  global pictstate,myWindow,moveMe,myNum
  play "click"
  if pictState = "true" then
    put "window "&myWindow &" into moveMe
  else

```

```

    answer "No window open on this card." with "OK"
    exit gphD
end if
put the loc of moveMe into myLoc
put the height of moveMe into myHeight
put item 1 of myLoc into xCord
put item 2 of myLoc into yCord
if yCord + myHeight > 384 then
    beep
    answer "This image can move no farther." with "OK"
    exit gphD
end if
add 93 to ycord

if ycord < 198 then
    set the loc of moveMe to xCord, 105
else
    if ycord < 291 then
        set the loc of moveMe to xCord, 198
    else
        if ycord < 384 then
            set the loc of moveMe to xCord, 291
        else
            set the loc of moveMe to xCord, 384
        end if
    end if
end if
end if
—reset location of window on card Script
put "put "&quote& myWindow &quote& " into myWindow"—
into line 13 of cd fld "cardScript" of cd "helpGuide"
put xcord& ","& ycord into windowLoc
put "put "&quote& windowLoc &quote& " into windowLoc" into—
line 14 of cd fld "cardScript" of cd "helpGuide"
newCdScript
end gphD

on gphR
global pictstate, myWindow, moveMe, myNum
play "click"
if pictState = "true" then
    put "window "&quote& myWindow &quote& into moveMe
else
    answer "No window open on this card." with "OK"
    exit gphR
end if
put the loc of moveMe into myLoc
put the width of moveMe into myWidth
put item 1 of myLoc into xCord
put item 2 of myLoc into yCord
if xCord + myWidth > 512 then
    beep
    answer "This image can move no farther." with "OK"
    exit gphR
end if
add 125 to xcord

if xcord < 262 then
    set the loc of moveMe to 137, yCord

```

**Scripts
for moving
Image Window
cont.**

```
else
  if xcord < 387 then
    set the loc of moveMe to 262, yCord
  else
    if xcord < 512 then
      set the loc of moveMe to 387, yCord
    else
      set the loc of moveMe to 512, yCord
    end if
  end if
end if
—reset location of window on card Script
put " put "&quote& myWindow &quote& " into myWindow"—
into line 13 of cd fld "cardScript" of cd "helpGuide"
put xcord& ","& ycord into windowLoc
put " put " &quote & windowLoc &quote& " into windowLoc" into—
line 14 of cd fld "cardScript" of cd "helpGuide"
newCdScript
end gphR

on gphL
global pictstate,myWindow,moveMe,myNum
play "click"
if pictState = "true" then
  put "window "&quote &myWindow &quote into moveMe
else
  answer "No window open on this card." with "OK"
  exit gphL
end if
put the loc of moveMe into myLoc
put the width of moveMe into myWidth
put item 1 of myLoc into xCord
put item 2 of myLoc into yCord
if xCord < 137 then
  beep
  answer "This image can move no farther." with "OK"
  exit gphL
end if
subtract 125 from xcord

if xcord < 137 then
  set the loc of moveMe to 12, yCord
else
  if xcord < 262 then
    set the loc of moveMe to 137, yCord
  else
    if xcord < 387 then
      set the loc of moveMe to 262, yCord
    else
      set the loc of moveMe to 387, yCord
    end if
  end if
end if
—reset location of window on card Script
put " put "&quote& myWindow &quote& " into myWindow"—
into line 13 of cd fld "cardScript" of cd "helpGuide"
put xcord& ","& ycord into windowLoc
put " put " &quote & windowLoc &quote& " into windowLoc" into—
```

```

line 14 of cd fld "cardScript" of cd "helpGuide"
newCdScript
end gphL

```

Handler to Update Card Script

```

on newCdScript
  global cdName
  put empty into scriptReader
  put the number of lines of cd fld "cardScript" of cd "helpGuide"~
  into fLines
  repeat with x = 1 to fLines
    put line x of cd fld "cardScript" of cd "helpGuide"~
    after scriptReader
  put return after scriptReader
  end repeat
  set the script of cd cdName to scriptReader
end newCdScript

```

```

—*****
— presenter Palette
—*****

```

Scripts for Presenter Palette

```

on editme
  global presenterGlobal,directorGlobal,myMode
  play "click"
  if not (directorGlobal contains "true") then
    palette "director","512,16"
  end if
  if presenterGlobal contains "true" then
    send "close" to window "presenter"
  end if
  put "edit" into bg fld "mode" of cd 1
  put "edit" into myMode
end editme

```

```

on printMe
  play "click"
  answer "Print which?" with "Text" or "To File" or "Cancel"
  if it = "Cancel" then exit printMe
  if it = "Text" then printText
  if it = "To File" then printFile
end printMe

```

```

on printFile
  put the short name of this stack && ".txt" into fileName
  ask file "Export to what file?" with fileName

```

```

  if it is empty then exit printFile —in case of cancel
  put it into fileName
  set the cursor to watch
  open file fileName
  repeat with counter = 1 to the number of cds - 3
    write cd field "title" of cd counter to file fileName
    write return to file fileName
    write return to file fileName
    write cd field "bodyText" of cd counter to file fileName
    write return to file fileName
    write return to file fileName
    write "Notes: " to file fileName
  end repeat

```

**Scripts for
Presenter
Palette
cont.**

```
write return to file fileName
write cd fld "TextEntry" of cd counter to file filename
write RETURN to file filename
write return to file fileName
write RETURN to file filename
end repeat

close file fileName
end printFile

on printText
  put empty into myMem
  lock screen
  set cursor to busy
  repeat with curCard = 1 to number of cards-3
    repeat with counter = 1 to 3
      show cd fld counter of cd curCard
      if cd fld counter of cd curCard is not empty then
        repeat with x = 1 to the number of lines in cd fld counter-
          of cd curCard
          put line x of cd fld counter of cd curCard after myMem
          put return after myMem
        end repeat
      end if
      hide cd fld "TextEntry" of cd curCard
    end repeat
    put return after myMem
  end repeat
  print myMem
end printText

on laserSetUp
  global playDisc
  play "click"
  answer "Set Laserdisc: " with "On" or "Off"
  if it = "On" then
    put "on" into playDisc
  else
    put "off" into playDisc
  end if
end laserSetUp
```

**Sample
Card Script
in Golden
Presentation**

```
** CARD #1: WilliamGolden *****
on openCard
  global cdName,pictState,myWindow
  global playDisc, bgColor>windowLoc
  put the short name of this cd into cdName
  if the hilite of cd btn "hiliter" is true then
    put "black" into bgColor
  else
    put "white" into bgColor
  end if
  —Programming for picture file
  put "true" into pictState
  if pictState = "true" then
    put "golden2" into myWindow
    put "12,105" into windowLoc
    picture myWindow,file,rect,false
```

```

    put item 1 of windowLoc into xLoc
    put item 2 of windowLoc into yLoc
    set loc of window myWindow to xLoc,yLoc
    set visible of window myWindow to true
end if
—Programming for laserdisc
if playDisc = "on" then
    put the number of this cd into lineNum
    put line lineNum of cd fld "discMem"~
    of cd "Laserdisc images" into frameNumber
    searchVideo frameNumber
    — put frameNumber
end if
end openCard

```

```

** CARD ** 1, BUTTON #1: hiliter
*****
on mouseUp
    set the hilite of me to true
    hide me
end mouseUp

```

```

** CARD #10 *****
on openCard
    global directorGlobal
    show cd fld "p1"
    show cd fld "pNum"
    if directorGlobal = "True" then
        set the loc of window "Director" to 6,116
    else
        palette "Director", "6,116"
    end if
end openCard

```

```

on closeCard
    global titleGlobal, subtitleGlobal
    global imageGlobal, presenterGlobal
    global directorGlobal, myMode

    —close extra palettes
    if titleGlobal contains "true" then
        send "close" to window "title"
    end if
    if subtitleGlobal contains "true" then
        send "close" to window "subtitle"
    end if
    if imageGlobal contains "true" then
        send "close" to window "image"
    end if
    if myMode contains "edit" then
        if presenterGlobal contains "true" then
            send "close" to window "presenter"
        end if
        if directorGlobal = "True" then
            set the loc of window "Director" to 512,16
        else

```

Card Script for Help Card in Golden Presentation

**Card Script
for Help Card
in Golden
Presentation
cont.**

```

    palette "Director", "512,16"
end if
else
    if directorGlobal contains "true" then
        send "close" to window "director"
    end if
    if presenterGlobal = "True" then
        set the loc of window "presenter" to 512,16
    else
        palette "presenter", "512,16"
    end if
end if
end closeCard

```

**Card Script
for Help Card
Field
in Golden
Presentation**

```

** CARD #10, FIELD #7: p1 *****
on mouseUp
    global ImageGlobal,PresenterGlobal
    global TitleGlobal,SubtitleGlobal

    get the clickLine
    select the clickLine
    if it is empty then exit mouseUp
    put word 2 of it into lineNum
    put line lineNum of cd fld "pNum" into myNum
    if (myNum < 2) or (myNum > 6) then
        exit mouseUp
        play boing
    end if
    put "p"&myNum into myPage
    lock screen
    hide cd fld "p1"
    hide cd fld "pNum"
    repeat with counter = 2 to 6
        put "p"&counter into theFld
        hide cd fld theFld
    end repeat
    show cd fld myPage

    —set up correct palettes
    if myNum = 2 then
        if imageGlobal contains "true" then
            send "close" to window "image"
        end if
        if presenterGlobal contains "true" then
            send "close" to window "presenter"
        end if
        palette "title", "6,223"
        palette "subtitle", "6,332"
    else
        if myNum = 3 then
            if titleGlobal contains "true" then
                send "close" to window "title"
            end if
            if subtitleGlobal contains "true" then
                send "close" to window "subtitle"
            end if
            palette "image", "6,223"
            palette "presenter", "6,273"
        end if
    end if
end mouseUp

```

**Card Script
for Help Card
Field
in Golden
Presentation
cont.**

```

else
  —close extra palettes
  if titleGlobal contains "true" then
    send "close" to window "title"
  end if
  if subtitleGlobal contains "true" then
    send "close" to window "subtitle"
  end if
  if imageGlobal contains "true" then
    send "close" to window "image"
  end if
  if presenterGlobal contains "true" then
    send "close" to window "presenter"
  end if
end if
end if
end if

```

end mouseUp

```

** CARD ** 10, BUTTON #1: Return
*****

```

```

on mouseUp
  go back
end mouseUp

```

```

** CARD ** 10, BUTTON #2: page2
*****

```

```

on mouseUp
  global directorGlobal,ImageGlobal
  global TitleGlobal,SubtitleGlobal
  global PresenterGlobal

```

```

  lock screen
  hide cd fld "p1"
  hide cd fld "pNum"
  show cd fld "p2"
  unlock screen

```

```

  —set up palettes
  if imageGlobal contains "true" then
    send "close" to window "Image"
  end if
  if presenterGlobal contains "true" then
    send "close" to window "presenter"
  end if
  palette "Title","6,223"
  Palette "Subtitle","6,332"
end mouseUp

```

```

** CARD ** 10, BUTTON #3: contents
*****

```

```

on mouseUp
  global ImageGlobal,PresenterGlobal
  global TitleGlobal,SubtitleGlobal

```

```

  —set up palettes
  if titleGlobal contains "true" then
    send "close" to window "title"

```

**Card Script
for Help Card
Buttons
in Golden
Presentation**

**Card Script
for Help Card
Buttons
in Golden
Presentation
cont.**

```
end if
if subtitleGlobal contains "true" then
  send "close" to window "subtitle"
end if
if imageGlobal contains "true" then
  send "close" to window "image"
end if
if presenterGlobal contains "true" then
  send "close" to window "presenter"
end if

lock screen
show cd fld "p1"
show cd fld "pNum"
unlock screen
end mouseUp

** CARD ** 10, BUTTON #4: page3
*****
on mouseUp
  global ImageGlobal,PresenterGlobal
  global TitleGlobal,SubtitleGlobal

  lock screen
  hide cd fld "p1"
  hide cd fld "pNum"
  hide cd fld "p2"
  show cd fld "p3"
  unlock screen

  —set up palettes
  if titleGlobal contains "true" then
    send "close" to window "title"
  end if
  if subtitleGlobal contains "true" then
    send "close" to window "subtitle"
  end if
  palette "Image","6,223"
  Palette "Presenter","6,273"
end mouseUp

** CARD ** 10, BUTTON #5: page4
*****
on mouseUp
  global ImageGlobal,PresenterGlobal
  global TitleGlobal,SubtitleGlobal

  —set up palettes
  if titleGlobal contains "true" then
    send "close" to window "title"
  end if
  if subtitleGlobal contains "true" then
    send "close" to window "subtitle"
  end if
  if imageGlobal contains "true" then
    send "close" to window "image"
  end if
  if presenterGlobal contains "true" then
```

**Card Script
for Help Card
Buttons
in Golden
Presentation
cont.**

```
    send "close" to window "presenter"
end if
```

```
lock screen
repeat with counter = 1 to 3
    put "p"&counter into myName
    hide cd fld myName
end repeat
hide cd fld "pNum"
show cd fld "p4"
unlock screen
end mouseUp
```

```
** CARD ** 10, BUTTON #6: page5
*****
```

```
on mouseUp
    global ImageGlobal,PresenterGlobal
    global TitleGlobal,SubtitleGlobal

    —set up palettes
    if titleGlobal contains "true" then
        send "close" to window "title"
    end if
    if subtitleGlobal contains "true" then
        send "close" to window "subtitle"
    end if
    if imageGlobal contains "true" then
        send "close" to window "image"
    end if
    if presenterGlobal contains "true" then
        send "close" to window "presenter"
    end if
```

```
lock screen
repeat with counter = 1 to 4
    put "p"&counter into myName
    hide cd fld myName
end repeat
hide cd fld "pNum"
show cd fld "p5"
unlock screen
end mouseUp
```

```
** CARD ** 10, BUTTON #7: page6
*****
```

```
on mouseUp
    global ImageGlobal,PresenterGlobal
    global TitleGlobal,SubtitleGlobal

    —set up palettes
    if titleGlobal contains "true" then
        send "close" to window "title"
    end if
    if subtitleGlobal contains "true" then
        send "close" to window "subtitle"
    end if
    if imageGlobal contains "true" then
        send "close" to window "image"
    end if
```

**Card Script
for Help Card
Buttons
in Golden
Presentation
cont.**

```

if presenterGlobal contains "true" then
    send "close" to window "presenter"
end if

lock screen
repeat with counter = 1 to 5
    put "p"&counter into myName
    hide cd fld myName
end repeat
hide cd fld "pNum"
show cd fld "p6"
unlock screen
end mouseUp

```

```

** CARD ** 10, BUTTON #8: showCdScript
*****
on mouseUp
    set the visible of cd fld 1 to not the visible-
    of cd fld 1
end mouseUp

```

**Card Script
for Move Card
in Golden
Presentation**

```

** CARD #11: MoveCard *****
on openCard
    global fldLines
    — create the list of card names
    put empty into cd fld "table of contents"
    —set scroll of cd fld "table of contents" to 0
    repeat with curCard = 1 to number of cards-3
        set cursor to busy
        get the short name of card curCard
        put it into line curCard of cd fld "table of contents"
    end repeat
    put the number of lines of cd fld "table of contents" into fldLines
end openCard

```

```

** CARD #11, FIELD #1: Table of Contents
*****
on mouseDown
    if the optionKey is down then
        createCardNameList — rebuild list
        exit mouseDown
    end if
    — go to the card associated with the click
    select the clickLine
    wait until the mouse is up
    if the mouseLoc is not within rect of me then
        select empty
        exit mouseDown
    end if
    put word 2 of the clickLine into theLine
    visual effect shrink to top
    go card theLine
end mouseDown

```

**Card Script
for Field on
Move Card
Field
in Golden
Presentation
cont.**

```
on createCardNameList
  — create the list of card names
  put empty into me
  if the style of me is "Scrolling" then set scroll of me to 0
  repeat with curCard = 1 to number of cards
    set cursor to busy
    get the short name of card curCard
    put it into line curCard of me
  end repeat
end createCardNameList
```

```
on newField
  — auto builds a new list when this field is pasted
  createCardNameList
end newField
```

```
on mouseUp
  —this handler traps mouseUp messages
  —that would otherwise move the palettes
end mouseUp
```

```
** CARD ** 11, BUTTON #2: Return
*****
on mouseUp
  global myTask
  if myTask contains "paste" then
    answer "Indicate the new location please." with "OK"
    exit mouseUp
  else
    go back
  end if
end mouseUp
```

**Button Script
on Move Card
in Golden
Presentation**

```
** CARD ** 11, BUTTON #3-26: line#
*****
on mouseUp
  global fldLines, myTask
  if myTask contains "paste" then exit mouseUp
  lock screen
  repeat 3 times
    set the hilite of me to true
    set the hilite of me to false
  end repeat

  answer "Perform what task?" with "rename" or "delete" or "move"
  put it into myTask

  —get the line number chosen
  put the short name of me into myNumber
  repeat 4 times
    delete char 1 of myNumber
  end repeat

  —get the name of the chosen card
  put line myNumber of cd fld "Table of Contents" into myName

  —job list for move card:
  if myTask contains "move" then
```

**Button Script
on Move Card
in Golden
Presentation
cont.**

```
answer "Move card "&myNumber&": "&quote&myName&quote~
with "Yes" or "No"
if it contains "No" then
    exit mouseUp
else
    lock screen
    go cd myNumber
    doMenu "Cut card"
    go back
    unlock screen
    Answer "Choose an arrow to show the new location." with "OK"
    put "paste" into myTask
end if
end if
—job list for rename card
if myTask contains "rename" then
    put quote & myName & quote into holder
    ask "New name for card: "&holder with myName
    if it is not empty then
        put it into newName
        set the name of cd myNumber to newName
        updateFldContents
    else
        exit mouseUp
    end if
end if
end if

—job list for delete card:
if myTask contains "delete" then
    answer "Delete card "&myNumber&": "&quote&myName&quote~
    with "Yes" or "No"
    if it contains "No" then
        exit mouseUp
    else
        lock screen
        go cd myName
        doMenu "Delete Card"
        go back
    end if
end if
end mouseUp
```

```

** CARD ** 11, BUTTON #27-51: paste1
*****

```

```

on mouseUp
  global myTask,fldLines
  if not (myTask contains "paste") then
    exit mouseUp
  else
    —get the new location
    put the short name of me into Pointer
    repeat 5 times
      delete char 1 of pointer
    end repeat

    —hilite the button
    repeat 6 times
      set the hilite of me to not the hilite of me
      wait 5
    end repeat

    —do the pasting of the card
    lock screen
    if Pointer > fldLines then
      go cd fldLines
      doMenu "Paste Card"
    else
      if pointer = 1 then
        —routine for moving card to first position
        go first
        doMenu "Paste Card"
        go first
        doMenu "Cut Card"
        doMenu "Paste Card"
      else
        go cd pointer
        go prev
        doMenu "Paste Card"
      end if
      go cd "moveCard"
    end if
    end if
    put "open" into myTask
  end mouseUp

```

Button Script on Move Card in Golden Presentation cont.

```

** CARD #12: Laserdisc Images *****

```

```

on opencard
  global fldLines,playdisc
  if playdisc = "on" then
    hide cd btn "off"
    show cd btn "on"
  else
    show cd btn "off"
    hide cd btn "on"
  end if
  set the scroll of cd fld "discDirectory" to 0

```

```

—enter the screen title into cd fld "table of contents"
put the number of cards -3 into fldLines

```

Card Script for Laserdisc Card in Golden Presentation

**Card Script
for Laserdisc
Card in Golden
Presentation
cont.**

```
put the short name of this stack into myStack
put empty into cd fld "table of contents"
repeat with curCard = 1 to number of cards-3
  set cursor to busy
  get the short name of card curCard
  put it into line curCard of cd fld "table of contents"
end repeat
put the number of lines of cd fld "table of contents" into fldLines
```

```
—enter None where fld is empty
repeat with x = 1 to fldLines
  put line x of cd fld "discMem" into holder
  if holder < 1 or holder > 30000 then
    put "none" into line x of cd fld "discMem"
  end if
end repeat
put fldLines + 1 into counter
repeat with x = counter to 24
  put " " into line x of cd fld "discMem"
end repeat
end openCard
```

```
** CARD #12, FIELD #1: Table of Contents
*****
```

```
on mouseUp
  select the clickLine
  get the clickLine
  put word 2 of it into screenNum
  put line 1 of cd fld "DiscPlace" into line screenNum-
  of cd fld "DiscMem"
end mouseUp
```

```
** CARD #12, FIELD #3: discDirectory
*****
```

```
on mouseUp
  global frameNumber
  get the clickLine
  —put the clickLine
  select the clickLine
  put word 2 of it into lineNum
  put line lineNum of cd fld "discNum" into frameNumber
  searchVideo frameNumber
  put frameNumber into cd fld "discPlace"
end mouseUp
```

```
** CARD #12, FIELD #4: DiscMem
*****
```

```
on mouseUp
  get the clickline
  select it
  put word 2 of it into linenum
  search line linenum of cd fld "discMem"
end mouseUp
```

```
** CARD #12, FIELD #6: HelpField
*****
```

```
on mouseUp
  lock screen
```

**Scripts
for Laserdisc
Card in Golden
Presentation
cont.**

```
hide me
unlock screen with visual dissolve
end mouseUp
```

```
** CARD #12, FIELD #7: players *****
on mouseUp
  global playerType — global used for player type throughout GDA Stacks
  put the value of the clickline into Holder
  put the clickline into selectLine
  select selectLine
  wait 20 ticks
  if Holder contains "cancel" then
    hide card field "players"
    hide cd button "cover"
  else
    if holder contains "none" then
      put empty into cd fld "currentPlayer"
    else
      put holder into playerType
      —setVideoPlayer playerType
      hide card field "players"
      put playerType into cd field "currentPlayer"
    end if
    wait 2 seconds
    hide cd button "cover"
    hide cd fld "players"
  end if
end mouseUp
```

```
** CARD ** 12, BUTTON #1: Return
*****
on mouseUp
  set the textAlign of me to left
  pop card
end mouseUp
```

```
** CARD ** 12, BUTTON #2: Back
*****
on mouseUp
  global frameNumber
  if line 1 of cd fld "DiscPlace" = empty then
    put 1 into frameNumber
    put frameNumber into cd fld "DiscPlace"
  else
    put line 1 of cd fld "DiscPlace" into myNumber
    subtract 1 from myNumber
    searchVideo myNumber
    — put MyNumber
    put myNumber into line 1 of cd fld "DiscPlace"
  end if
end mouseUp
```

```
** CARD ** 12, BUTTON #3: Forward
*****
on mouseUp
  global frameNumber
  if line 1 of cd fld "DiscPlace" = empty then
```

**Card Script
for Laserdisc
Card in Golden
Presentation
cont.**

```
    put 1 into frameNumber
    put frameNumber into cd fld "DiscPlace"
else
    put line 1 of cd fld "DiscPlace" into myNumber
    add 1 to myNumber
    searchVideo myNumber
    —put MyNumber
    put myNumber into line 1 of cd fld "DiscPlace"
end if
end mouseUp
```

```
** CARD ** 12, BUTTON #4: Help
*****
```

```
on mouseUp
    lock screen
    set the visible of cd field "HelpField" to~
    not the visible of cd fld "HelpField"
    unlock screen with visual dissolve
end mouseUp
```

```
** CARD ** 12, BUTTON #5 *****
```

```
on mouseUp
    global fldLines
    repeat with x = 1 to fldLines
        set hilite of cd button "slide show" to true
        if not (line x of cd fld "discMem" contains "none") then
            select line x of cd fld "discMem"
            searchVideo (line x of cd fld "discMem")
            put "displaying " & line x of cd fld "discMem"
            wait 130 ticks
        end if

        if the mouseClicked is true then
            set hilite of cd button "slide show" to false
            exit repeat
        end if
    end repeat
    set hilite of cd button "slide show" to false
end mouseUp
```

```
** CARD ** 12, BUTTON #6-29: line1
*****
```

```
on mouseUp
    get the short name of me
    repeat 4 times
        delete char 1 of it
    end repeat
    put it into screenNum
    put line 1 of cd fld "DiscPlace" into line screenNum~
    of cd fld "DiscMem"
end mouseUp
```

**Card Script
for Laserdisc
Card in Golden
Presentation
cont.**

```
** CARD ** 12, BUTTON #30: Clear
*****
on mouseUp
  Answer "Clear all selections?" with "Yes" or "No"
  if it = "yes" then
    put empty into cd fld "DiscMem"
  else
    put empty into line 1 of cd fld "DiscPlace"
  end if
end mouseUp
```

```
** CARD ** 12, BUTTON #35: Set Player
*****
```

```
—Set Player button Script
on mouseUp
  global typeOfVideo
  global playerType
  set hilite of target to true
  put playerType~
  into cd field "currentPlayer"
  show card field "players"
  show cd field "currentPlayer"
  —show cd button cover
end mouseUp
```

```
** CARD ** 12, BUTTON #36: off *****
```

```
on mouseUp
  global playDisc
  hide me
  show cd btn "on"
  put "on" into playDisc
end mouseUp
```

```
** CARD ** 12, BUTTON #37: on *****
```

```
on mouseUp
  global playDisc
  hide me
  show cd btn "off"
  put "off" into playDisc
end mouseUp
```

Appendix H: Glossary

analog: Varying smoothly and continuously over a range, rather than changing in discrete jumps. For example, the signal recorded on videotape is an analog signal because the signal is continuously variable. The information stored on a computer disk is digital because the information is stored as discrete binary information.

analog

animation: A series of static images which, when presented in rapid succession, create the illusion of movement. The images can be recorded in analog form on film or videotape, or digitally on tape or disk.

animation

application: A computer program that performs a specific purpose, such as word processing, database management, graphics, sound, or animation.

application

anti-aliasing: smoothing edges created with painting, selection, or type tools.

anti-aliasing

ASCII: (American Standard Code for Information Interchange) ASCII is a worldwide standard for representing text as numbers, rather than as bits.

ASCII

Authoring Software: Any program that is used to create, read from peripheral devices, combine, edit, and produce multimedia or hypermedia presentations; ideal for specifying relationships between segments or user-controlled events; examples are HyperCard, SuperCard, Plus, or Authorware.

**authoring
software**

CD-I: A computerized CD player built by Philips that plugs into a television set instead of using a computer screen.

CD-I

CD-ROM: Compact Disc - Read Only Memory is based on CD audio; information is stored in binary digital format. Compact discs that can be played on a personal computer. The first technology to exploit the huge storage capacity of CDs, it requires different discs for different brands of computers.

CD-ROM

CDTV: Commodore's version of CD-I, marketed earlier but with a more limited selection of software.

CDTV

Digitize - convert analog signal into digital information (pixels) recording color (chrominance) and gray scale (luminance).

digitize

GDA - The Graphic Design Archive.

GDA

Generational loss: reduction in picture quality resulting from copying video signals for video editing and distribution.

**generational
loss**

genlock: (short for synchronization generator locks), a device that synchronizes the computer's output with the video signal. This requires the addition of a videographics card to a Macintosh II.

genlock

GUI	GUI: Graphical User Interface.
HyperCard	HyperCard: a programming environment offering non-specialists easy scripting and simultaneous work with text, image, and sound. This program is bundled with the purchase of the Apple Macintosh computer.
icon	icon: a graphic image that represents an object, a concept, or a message.
interactive	interactive: the process of allowing the user to be in control of the choices made within a computer program.
interlaced video	interlaced video: the process of scanning video frames in two passes, with each pass painting every other line of the frame onto the screen. NTSC's 525-line from scans in two fields of 262.5 lines each that take 1/60 second to paint. Thus, each frame takes 1/30 second to paint. see noninterlaced video.
Kaleida	Kaleida: A new joint venture by Apple and IBM to develop their version of the multimedia computer of the future. First product due in the mid-'90s.
Macintalk	Macintalk: a speech-generating program that allows the Macintosh to convert typewritten text to spoken English.
MacRecorder	MacRecorder - simple box that digitizes audio (voice, microphone, radio, stereo) allowing audio to be recorded, edited, and added to Macintosh programs.
MacroMind Director	MacroMind Director - Macintosh program that allows you to create presentations and/or movie segments. Movie segments can be turned into stand alone presentations or used in combination with other programs, such as HyperCard.
mapping	Mapping: taking a two-dimensional surface texture (wood, marble, etc.) and wrapping it onto a three-dimensional surface.
MIDI	MIDI: A communication protocol for the synchronization and control of MIDI musical instruments and other devices. It is an acronym for Musical Instrument Digital Interface.
MPC	MPC: A personal computer with a CD drive built in. More than 70 companies, led by Microsoft and Tandy, are producing MPC hardware and software.
noninterlaced video	noninterlaced video: the process of scanning complete video frames in one pass. This usually produces higher image quality than interlaced video produces. The Mac's normal display is noninterlaced.
NTSC	NTSC: National Television Standards Committee, which defined the 525-line, 30-frame-per-second TV standard currently used in the U.S., Canada, Mexico, Japan, and a few other countries.
PAL	PAL: Phase Alteration Line System, the system of broadcast video display used in Europe.

<p>persistance of vision - refers to the mind's inability to separate images from each other when shown at a certain frequency in sequence to create the effect of motion.</p>	<p>persistance of vision</p>
<p>Persuasion - From Aldus software, this program allows the user to create a presentation on the computer, which can be viewed on the computer, in slide format, or transparencies. It also creates handouts and outlines to accompany the presentation.</p>	<p>Persuasion</p>
<p>pixel - the smallest visible element on a video screen; from "picture element". On the Macintosh it is 1/72 of an inch square.</p>	<p>pixel</p>
<p>presentation: Any production from a single slide to a full animated slide show with movies and sound.</p>	<p>presentation</p>
<p>Resource: one of two forks that make up every Macintosh file. (The other fork is called the Data fork.) All externals, fonts, sounds, and icons reside in a file's resource fork.</p>	<p>resource</p>
<p>RGB: method of transmitting video signals that feeds the basic red, green, and blue channels over separate wires. This provides the highest-quality video signal and is the native format for most computer equipment.</p>	<p>RGB</p>
<p>Sampling: the digitization of analog music sources by sampling the source at frequent enough intervals to (hopefully) fool the human ear.</p>	<p>sampling</p>
<p>SANE: Acronym for Standard Apple Numerics Environment, a set of arithmetic routines common to all Apple Computers that guarantee the accuracy of computations.</p>	<p>SANE</p>
<p>SECAM: An acronym for Sequential Color with Memory, the system of broadcast video display in the Soviet Union and France.</p>	<p>SECAM</p>
<p>sequence: A series of pictures or notes. In animation, a sequence of art is any series of frames in one channel of the score. In music, it is a series of notes arranged on a MIDI sequencer and played on a MIDI instrument.</p>	<p>sequence</p>
<p>Sprites: in animation, small elements that move in the foreground, existing independently of the background.</p>	<p>sprites</p>
<p>SuperCard: a multimedia authoring environment, developed for creating stand-alone Macintosh applications.</p>	<p>SuperCard</p>
<p>transition: A visual effect that occurs between two different frames (in animation) or cards (in HyperCard).</p>	<p>transition</p>
<p>VHS: a 1/2-inch video system developed by Matsushita that has become the predominant home format.</p>	<p>VHS</p>
<p>videodisc: While they look like large CDs, they can store sound, text, still pictures, and full motion video. In addition they work well for fast random searches, making them ideal for multimedia.</p>	<p>videodisc</p>

WYSIWYG

WYSIWYG: Acronym for "What you see is what you get," meaning that what you see on the screen is what you get on paper when you print a document.

XCMD

XCMD: The resource type for, and popular name given to, an external command (used in HyperCard and MacroMind Director.)

XFCN

XFCN: The resource type for, and popular name given to, an external function (used in HyperCard and MacroMind Director.)

Appendix I: Bibliography

- Ambron, SueAnn. Learning with Interactive Multimedia.
Redmond, Washington: Microsoft Press 1990.
- Ambron, SueAnn and Kristina Hooper. Interactive Multimedia.
Redmond, Washington: Apple Computer Inc. 1988.
- Apple Computer, Inc. HyperCard Stack Design Guideline.
Reading, Massachusetts: Addison-Wesley Publishing
Co., Inc. 1989.
- Apple Computer, Inc. HyperCard: Script Language Guide.
Santa Clara, CA: Claris. 1989-1990.
- Beekman, George. "Insights on HyperCard 2." MacWorld.
September 1991. pp. 297-300.
- Bonsiepe, Gui. "Interpretations of Human User Interface."
Visible Language. Vol XXIV. 3/4. pp. 262-285.
- "Colorizing Your Palettes." Inside HyperCard. January 1992.
Vol. 2, No. 1. pp. 7-8.
- Coupland, Ken. "Breaking into Pictures." HOW. July/August,
Vol. VI, No. 5. pp. 62 - 69.
- Durlak, Jerry. "The Glenn Gould Profile." Minds in Motion.
Fall 1989. pp. 7-18.
- Elmer-Dewitt, Philip. "The World on a Screen." Time. October
21, 1991. pp. 80-82.
- Gates, Bill. "Gates' Vision: Putting Archives to Use."
USA Today. October 9, 1991. p. 8B.
- Goodman, Danny. The Complete HyperCard 2.0 Handbook.
New York: Bantam Books. 1990.
- Heid, Jim. "Getting Started with Multimedia". Macworld.
May 1991. pp. 225-232.
- Himes, Andrew and Craig Ragland. Inside SuperCard.
Redmond, Washington: Microsoft Press. 1990.
- Hoffmann, Isabel de Correa. "HyperCard in an Artistic
Environment." Minds in Motion. Fall 1988. pp. 27-31.
- "Interactive Media and Hypercard" from Minds in Motion,
Spring 1989.

- Kerr, Stephen T. "Instructional Text: the Transition from Page to Screen." Visible Language. Autumn 1986. pp. 368-392.
- Kool, Richard and David Denning. "The Jason Project: Creation and Evaluation." Minds in Motion. Fall 1989. pp. 45-57.
- Lai, Patrick and Udi Manber. "Flying Through Hypertext." Hypertext '91 Proceedings. December 1991. pp. 123-132.
- Lehrman, Paul D. & Tim Tully. "Catch a Wave: Digital Audio." MacUser. October 1991. pp. 94-103.
- "MacroMind Director: Interactivity Manual." by MacroMind, Inc., April 1990.
- Mager, Robert F. Preparing Instructional Objectives. Belmont, California: Fearon Publishers. 1962.
- Malin, Stuart. "Hypermedia and Design." Design Department. March 14, 1990. pp. 6-11.
- Martin, Teresa A. "Multimedia Grapples with Interface Design." Macweek. October 1990. pp. 34-40.
- McQuillin, Lon. "Authoring, Modeling, and Animation." MacUser. Feb. 1991. pp. 52-68 in Buyer's Guide Supplement.
- McQuillin, Lon. "Multimedia and Video." MacUser. Feb. 1991. pp. 4-32 in Buyer's Guide Supplement.
- McQuillin, Lon and Ken Gruberman. "Multimedia and Audio." MacUser. February 1, 1991. pp. 38-50 in Buyer's Guide Supplement.
- Parker, Elisabeth Betz. "The Library of Congress Non-Print Optical Disk Pilot Program." Information Technology and Libraries. Dec 85. pp. 289-299.
- Petersen, Toni and Pat Molholt (ed.). Beyond the Book: Extending MARC for Subject Access. Boston: G. K. Hall & Co. 1990.
- Remington, R. Roger and Barbara J. Hodik. Nine Pioneers in American Graphic Design. Cambridge: MIT Press. 1989.
- Rubin, Tony. User Interface Design for Computer Systems. John Wiley and Sons. 1988.
- Stam, Deidre C. "The Quest for a Code." Beyond the Book. (Edited by Toni Petersen and Pat Molholt. Boston: G.K.& Hall. 1990. pp. 117-143.
- Sosinsky, Barrie. Beyond the Desktop: Tools and Technology for Computer Publishing. New York: Bantam Books. 1991.

- Strothman, Jim. "Columbus, New Products, Fill IBM Sails." Computer Pictures. October/November 1991. pp. 64-65.
- Swain, Michael. "Authorware Professional". MacUser. August 1, 1991. pp. 54-55.
- Thompson, Keith. "Media and the Message." PC Publishing and Presentations. August/September 1991. pp. 24-26.
- The Visual Craft of William Golden. Edited by Cipe Peneles Golden, Kurt Weihs, & Robert Strunsky. New York: George Braziller, Inc. 1962.
- Walker, Edwin G. "Graphic Design Archive 3.1: The Corporate Identity Module." Thesis Report. (Rochester Institute of Technology Archives, 1991).
- Weiman, Liza. "The Secret of the SuperStacks." MacWorld. April 1991. pp. 124-129.
- West, Nick. "MultiMedia Design Tools." MacWorld. November 1991. pp. 194-201.
- Wilson, Stephen. Multimedia Design with HyperCard. Englewood Cliffs, NJ: Prentice Hall. 1991.
- Winkler, Dan and Scot Kamins. HyperTalk 2.0: the Book. New York: Bantam Books. 1990.
- Yi, Paul. "The Mac/TV Connection: Desktop Video." MacUser. July 1991. pp. 124-134.