DOCUMENT RESUME

ED 352 038 IR 054 260

| | |
|---|---|
| TITLE | Hypermedia and Research Methodology: An Associative Research Model. Final Report. |
| INSTITUTION | City Univ. of New York, N.Y. Bernard Baruch Coll. |
| SPONS AGENCY | Office of Educational Research and Improvement (ED), Washington, DC. |
| PUB DATE | [91] |
| CONTRACT | R039A90033 |
| NOTE | 67p. |
| PUB TYPE | Guides - Non-Classroom Use (055) -- Reports - Evaluative/Feasibility (142) |
| EDRS PRICE | MF01/PC03 Plus Postage. |
| DESCRIPTORS | *Authoring Aids (Programing); College Libraries; *College Students; *Computer Software Development; Higher Education; *Hypermedia; Individualized Instruction; Instructional Effectiveness; *Library Instruction; *Research Skills |
| IDENTIFIERS | *HyperCard |

ABSTRACT

The purpose of the project described in this three-part report was to test the value of hypermedia for library instruction. The project included the development of a hypermedia program built around a marketing topic; implementation and testing of the program with one group of students assigned to use the new program; evaluation of the project by an outside evaluator; and a follow-up phase that included project modifications. Project findings included: (1) hypermedia at the low end of the technological media is effective for teaching some problem-solving research skills; (2) HyperCard serves as effective authorware for the content-oriented producer who has minimal prior knowledge of authoring software, but new content-oriented producers can produce high quality programs in much less time; (3) many students like the independence offered by these kinds of learning environments; (4) non-native speakers of English especially appreciated this learning environment because it was self-paced; and (5) hypermedia was responsive to research-oriented curricular needs in the library setting. The second part presents the report of the project's outside evaluator, Martin Kesselman. Based on observations, evaluation of the HyperCard program developed for the project, testing of the program by students and student reports, this evaluation found that hypermedia technology was an effective approach for library instruction. Several suggestions for improvements are included in the consultant's report. A manual for creating HyperCard programs for library instruction by Madeline Andrea Ford, which makes up the third part of the report, covers hardware, software, steps for designing the HyperCard program, selection of graphics, and creating stacks. (KRN)

**Project Title: HEA II-B Library Research and Demonstration (Pr/Award Number R039A90033)**

# Hypermedia and Research Methodology: An Associative Research Model Final Report

This report consists of three parts:

- The Project Final Report ( Eleanor Langstaff, Director of Project)
- The Evaluative Report ( Martin Kesselman, Coordinator, Electronic Reference/Instructional Services, Rutgers University Library of Science and Medicine)
- Creating HyperCard Programs for Library Instruction; a Manual (Madeline Ford, Project Associate)

Project Director:
Eleanor Langstaff

Project Associates:
Ida B. Lowe
Rona Ostrow

Baruch College, Box 317
City University of New York
17 Lexington Avenue
New York, NY 10010

# Part One: The Final Report

## Table of Contents

**The contents of this report were developed under a grant from the Department of Education. However, those contents do not necessarily represent the policy of the Department of Education and you should not assume endorsement by the Federal Government.**

# Part One: The Final Report

## Table of Contents

## Objectives of the Project and Summary of Results, followed by a discussion.

During the course of the project we expected to test the hypotheses listed below by means of the hypermedia program, HyperCard. HyperCard is a database template that comes with a selection of materials and tools to be used as needed. Basic to it are the "stacks," or files, and the "cards," or records. Any card can be linked to any other card in its stack, or to any other stack. Thus it is template for an undetermined relational database. The producer supplies the data and selects the links as needed to create the database and its relationships. As a medium of instruction it is especially useful for individualized learning since it is self-paced, and can be operated in linear fashion and by branching. Since the branches can be either enriching or remedial, the same program can be used by students with a variety of needs. In a library setting, which traditionally teaches one-on-one, it is more like personal instruction because of this branching capability. Moreover, since library materials tend to have frequent detail changes, its ease of updating is invaluable. Unlike print sources retained for historical value, outdated material in the HyperCard programs no longer serve as a trap for the unwitting student.

HyperCard can be the driver for video tape and laser disk visuals with or without sound. Our version omitted this kind of package in favor of something that could be used with less sophisticated equipment, so that it could be widely disseminated and used independently. This low-end program introduced students to print and computer-stored market research and site selection resources directly related to a case study "Mass to Class: Retargeting and Relocating a Retail Bookstore."

### Hypotheses

•Hypermedia is an effective tool for library user instruction.

•A widely available, easy to learn authoring program, HyperCard, can be used to produce effective instructional programs .

•Autonomous learning is encouraged by the use of this kind of material.

•Learners of differing abilities can use this type of program effectively.

•Hypermedia is especially responsive to project oriented curricular needs.

### Summary of Findings:

• We found that hypermedia at the low end of the technological range, for instance, HyperCard stacks, is effective for teaching some problem-solving research skills including secondary analysis of data by independent, on -demand use of the programs by students.

• We found that HyperCard serves as an effective authorware for the content-oriented producer who has minimal prior knowledge of authoring software, but that this self-learning mode for the authorware was time intensive. We found that new content-oriented producers working independently but with the advice of the first-generation producers produced high quality programs in half the time, and that this time was halved yet again for the third generation, so that the preparation time is identical with other modes of library instruction, such as the development of transparencies and handouts, while technical quality is higher and the program can routinely be kept current.

• We found that many students liked the independence offered in this learning environment. On a less positive note, others felt that the computer enhanced or guaranteed the authoritativeness of the information so provided, while others raced through the linear program without any desire to explore the branches that supplied depth. Subsequent versions of the program built in brakes and self- tests to encourage students to add depth to their knowledge.

• We found that this environment was especially appreciated by non-native speakers of English since it is self-paced. Further, this group of students tended to explore the branches of the program, so that they ended with a better than average grasp of the material.

• We found that hypermedia is responsive to research-oriented curricular needs in the library setting. Low-end hypermedia trades off ease of replication and unsupervised use for limits in file sizes, since the graphics that make the program so effective take up much disk space. One solution was to break the program out into components, although this hampered full branching.

• Finally, we found that results warranted training three more producers and producing two more stacks to date, one for library orientation and one on the use of the all-University online catalog; both of these have been required experiences for the Freshman Seminar participants.

## Discussion

- Hypermedia is an effective tool for library user instruction.

A major frustration of library user instruction is caused by the logistical problems associated with reaching all students who have a specific need. At Baruch, despite the fact that each semester the instructional division reaches 4,000 students in course-related lectures, and approximately 200 students in credit courses, a great portion of the Baruch population--nearly 20,000 students and faculty--are not reached and instructed in this most crucial of skills, information handling. Hypermedia offers the means to teach large numbers the use of information in research, decision-making, and problem-solving. The alternative, radically decreasing the student-librarian ratio, is too heavy in manpower costs to be an option in academic budgets. Because of its interactive and graphics features hypermedia offers an attractive alternative to one-on-one instruction.

- We found that Hypermedia is an effective tool for library user instruction when the program is related to the informational needs of the user. We further found that the more comfortable students are with any form of information technology the more effective is their learning from it, and that it should be made clear to students what benefits they can expect from the hypermedia program. ***Mass to Class*** suggested to several students by its introduction that it was a simulation package, containing all necessary data. This raised expectations that were not realized. In later programs (listed in Followup) we set goals more directly and incorporated self-tests after each part . Post-tests showed significant improvement in learning with hypermedia.

- A widely available, easy to learn authoring program, HyperCard, can be used to produce effective instructional programs .

Changes in the economy of New York City have made the notion of 'mparting information skills less exclusively academic and increasingly more vocational/occupational in orientation. The U.S. economy in general has been transformed from an industrial to an information-based economy where even entry level positions require technical information-handling skills that too many of the city's high school and university graduates do not possess. This development affects all post-secondary schools and challenges campus librarians to harness technology to teach information-handling skills to whole student populations. This situation affected the choice of our first research HyperCard program in retail marketing.

We found that the use of a simple authoring system with templates and libraries of graphics makes it possible to make material relevant and graphically attractive with a reasonable expenditure of time and effort on the part of the librarian developer. We found that planning and shameless borrowing of ideas and techniques affected efficiency the most.

The essence of how we learned to make instructional HyperCard stacks is detailed in *Creating HyperCard Programs for Library Instruction: a Beginner's Manual* by project associate Madeleine Ford. We also profited from the training workshops held in conjunction with

Macademia '89 at Brown University.

- Autonomous learning is encouraged by the use of this kind of material.

The use of hypermedia for bibliographic instruction is a subject of growing interest at conferences such as LITA, ASIS, and ALA. In commenting on traditional methods of bibliographic instruction, James Cavanaugh, SUNY/Stonybrook librarian, stated at the recent LITA conference in Boston, "A linear train of thought reflects the author's (in this case, the librarian's) brain patterns, not the reader's." Interest is high outside the library field as well. At the Modern Language Association's 1989 meeting, no fewer than four meetings were held to discuss the cognitive and affective qualities of hypermedia.

•We found that autonomous learning is effective when the student is comfortable in the learning environment and equally at home with the technology. The long range interested we have in developing materials now is that a full range of programs will be in place for the opening of our new state-of-the-art library in 1993. Baruch's new library will be technology-based and we intend to make hypermedia programs available throughout the facility for student and faculty use. Although intended to improve service to our college constituency, the prototype package and subsequent packages are also available as models for the region and nation in demonstrations. To date we have demonstrated them once nationally and four times regionally.

- Learners of differing abilities can use this type of program effectively.

There are two ways in which this program can be used for learners of differing abilities. One is internal and self-directed in which the student decides whether or not to avail herself of additional information. In some cases the information is enriching and in others offers explanations that may be known to some users and therefore are hidden until asked for. An example is the glossary of retail marketing terms that is accessed in Mass to Class from any card. Later versions of these programs will utilize the sound capability to have glossary words spoken aloud as well as explained.

Within the framework of the program, we have found that with minor changes and additions, we were able to convert material for use with different age levels. We used, for instance two stacks (modules) from the Online Searching module to create ***Can You think Boolean?*** Based on the sample unit from the State Education Department's model syllabus for library instruction for Grade 8, it teaches simple Boolean searching as well as the use of a print index, ***Readers Guide***, and a reference tool about sports.

•We found that modification of programs for groups was more immediately effective than building in the appropriate links to reach all levels of student. Students in our test group tended to skip the additional information in favor of completing the assignment quickly. We will continue to design multi-level programs, for more effective independent use while improving the design to tempt students into more thorough use of the programs. We think that in the new library, where workstations will be scattered in many subject or activity oriented areas, our experience will result in the availability of a wide variety of effective independently learned materials.

Another factor that will raise quality in future productions is the lowered cost of the medium- to high-end of hypermedia. For those programs that must be used to cover complex concepts related to library research in a field, color, sound and animation will foster interest.

•Hypermedia is especially responsive to project oriented curricular needs.

The most important tenet of Baruch's philosophy of research instruction is that, since information overload is the norm, a student must always be in a situation of selecting from a variety of usable sources, and never reduced to selecting "enough." Simulation programs have long been used to develop judgment. We found that the closer the approximation to the real situation, including actual usable data, the more satisfied the students were with the experience, and the better judgment they exercised. Low end limitations included incomplete simulation.

## 3. Approach

### a. Plan of action

The Start-up Phase established the project committee (Eleanor Langstaff, project director; Ida Lowe, technical advisor; Rona Ostrow, bibliographic advisor; and two members of the Baruch School of Business faculty). This group selected a marketing topic for the hypermedia program. The project director then recruited, hired and trained technical support staff to help create the programs , and blocked out the testing procedure. In close cooperation with Ida Lowe, the technical expert, she ordered equipment.

The equipment ordered included a Macintosh IIx, laser printer, scanner and external 5 1/4" drive, and modem. Three Mac SE 30s were ordered as student work stations. Software included a wide variety of graphics packages, wordprocessing, animation and telecommunications software and their associated supplies.

The HyperCard Stack Production Phase saw the creation of nodes and links of instructional data sources. During this period Rona Ostrow identified and gathered both standard library resources and materials available only at the site of the case study, Cincinnati, Ohio. Based on the case study written by David Rachman, situational scenarios were developed. Finally the parts were merged into "Mass to Class: Relocating and Retargeting a Retail Bookstore."

During this time, we developed an animated and spoken version of the case study, which was found to be very effective when we presented it to marketing department members. Over time and frequent use, it tended to malfunction and was regretfully eliminated in the beta version which substituted a cartoon series for th^ :peaking actc . The cartoon characters introduced the case study and led into a presentation of various resources and their uses, both in relation to the data needed fro the project but a'so as a general overview.

The Implementation and Testing Phase had to be modified from the original intent because unforeseeable conditions within the School of Business reduced our sample to a size not consistent with statistical validity. The experimental section was divided into two groups. Students in the hypermedia group were given very brief instruction in the use of the program and then instructed to begin their research. The rationale was that the use of the program was self-explanatory and learning to use the Macintosh computer much easier than the IBMs the students were used to. Students in the control group received a traditional bibliographic instruction session on how to do research in marketing presented by the most effective of the bibliographic instructors and told to begin their research. The term reports of both groups were evaluated in terms of their use of research tools and strategies as well as their overall merits as decision-making documents. The students were also canvassed about what they added to their knowledge of research principles and practices as well as their overall attitude toward library research. The results are discussed by Martin Kesselman in part two of this report, that of the external evaluator. One result of this part of the study was that the submerged computer anxiety of the students was a factor in how they used the program. The expected sense of exploring and testing the computer's capabilities did not emerge. As a result of this experience, far more time is spent introducing the student users to the equipment, no matter how easy to operate.

The Followup Phase (to September 1990) included training one new producer, and modifying existing programs in light of the external evaluator's recommendations. For example, the white on black design of the menu card, which had been based on the very chic menu card of a commercial producer was changed to black on white and made more direct. Students immediately oriented themselves to the program more rapidly. This fast orientation is needed for full utilization of the program.

We also continue to give demonstrations, to academic and New York City High School Librarians and also to the Baruch College Faculty in 1991. Our greatest outreach, however, is installing a Library Help Folder in all available Macintosh computers on campus. With some publicity in the school newspaper, *The Ticker*, and by flyers, this makes this form of library instruction on demand available about 12 hours a day

The Followup Phase (ongoing)

That this project has interest for students was clear when the student newspaper, *The Ticker,* did a followup article a year later. (see Appendix) We are using new producers to develop new programs to reach new target groups. We are cooperating with the HyperCard Library Instruction Program for some standards. We do continue, however, to hold to the Baruch Library Instruction concept that programs developed be justifiable in both content and context; not "how to use a research tool" but "how to get information on a topic." For our particular purposes, while the excellent material from HLIP is appropriate for use of a particular tool, we prefer broader approaches. Material now in production is an overview of *The New York Times* as an information source, with reference to its editorial policy, arrangement, and index.

Another part of the follow up phase is to upgrade our programs and prepare for a higher level of hypermedia. Two projects that will use the higher levels of hypermedia are now in the planning stages. One is a program to serve as a refresher course for students taking science courses after some time away from this material. Another is a module in the development of information technology with emphasis on the multicultural aspects.

b. Chronological schedule of accomplishments and target dates

**September 1989**

- Established project committee.(Langstaff, Lowe, Ostrow)
- Selected topic for the hypermedia program within the area of marketing and in consultation with cooperating members of marketing department faculty. (Rachman, Fabes, Ostrow)
- Hired and began training of clerical and technical assistants. (Langstaff)
- Ordered equipment. (Langstaff, Lowe)

**September 1989-December 1989**

- Created the basic HyperCard stack for the marketing package and developed the package in consultation with the project committee. The case study, Mass to Class: Relocating and Retargeting a Retail Bookstore (in Cincinnati) was developed in conjunction with marketing faculty and then broken down into instructional nodes. (Langstaff, Ford) The bibliographic materials were produced. (Ostrow)

**January- -March 1990**

- Completion of first version of HyperCard associative learning module in marketing.
- 2 HyperCard workstations are made available for student use in reference areas of library and two installed in the online classroom.

## April 1990

Bibliographic instruction session for control group. (Pollard) - Students in both groups begin research papers.

- Beta test of instructional package in library reference area. HyperCard program to be loaded on 2 McIntosh computers in the reference area and are available for use only by test participants. Point-of-use evaluative materials are available for user reactions and suggestions. (Reference staff)

. - Examined and evaluated point-of-use responses. Responses tended to find the technology a barrier, and the access (their names had to checked against a list) a bore. Students using the program in the computer lab tended to feel that they had learned more than those using it in the library reference areas.

## May 1990

- Research papers submitted.
- Analyzed results of second research papers for evidence of research strategy, quality and quantity of sources used.- Post-test of students in both groups. As the instructor at the last moment has students work on the project in groups, each group pooled resources, whether in the control groups or in the hypermedia groups.
- Comparison of post-test results of HyperCard group and control group showed a similarity in both sectors, probably because of group work.
- Demonstrated module for New York City High School Librarians Annual Conference.(Langstaff, Ford)

**Summer 1990**

- Completed evaluation of results. - Prepare paper for publication on the results of the research. - Outside evaluator examines project and research results.(Langstaff, Kesselman)

-Presented "Mass to Class" as invited speaker in Macademia '90 national conference, Rochester, New York.(Ida Lowe)

- Program, as revised, made generally available to students in the reference area. Later moved to Macintosh Lab for greater accessibility.

d. Organizations, cooperators, consultants, and key individuals who worked on this project

1. Baruch College CUNY

The project was undertaken by Baruch College, City University of New York whose Library Department of the College has had sixteen years of experience in bibliographic instruction at both the course-related and credit-course levels. Members of the faculty of the Marketing Department, Baruch College School of Business and Public Administration, specifically Professors David Rachman and Keith Fabes provided the subject expertise in the field of marketing for the project. They produced the scenarios and determined evaluation points for the students' research papers from a substantive point of view.

c. Outside Evaluator

Martin Kesselman, Coordinator of Library Instruction, Library of Science and Medicine, Rutgers University, and developer of a HyperCard instructional program for users of the Rutgers Library of Science and Medicine.

d. Biographical sketch of program director and list of key personnel

Eleanor Langstaff, the project director, is coordinator of the credit-bearing courses. She has developed the information studies program, including curriculum development and the approval of six courses through state level. Further, she has been teaching in the program from its beginning, has co-authored the first textbook and the two texts currently in use, as well as two other books and other materials. Her dissertation, in progress, is based on research in semiotics. She has had communications training and experience as a Foreign Service Officer with the United States Information Agency. She has an M.A. in English from Fordham University, an M.S. in L.S. from Catholic University of America, and is currently completing her PhD in English at the City University of New York. The director was released from 6 credit hours of teaching annually to devote her time to fulfilling her functions as overall manager, especially in regard to the design of the materials. She was responsible for the overall pedagogic content of the materials developed.

Ida Lowe. the technical advisor to the project, has extensive administrative and teaching experience. She recently wrote and managed a $94,000 CSTEP grant, has developed and taught the online information retrieval courses, served as Acting Assistant Dean for Research and Development, and is currently Coordinator for Systems and Technical Services. She has an MBA in Computer Methodology, MSLS, and an MA in Political Science. Professor Lowe shared with the director the responsibility of sharing our findings with the national community, participating as an invited speaker at Macademia '90.

Rona Ostrow,. business school liaison and bibliographic advisor for the project, selected the materials for the program. At the time of the project she was the Director of the Graduate Business Resource Center. She has authored the workbook used in the library's business research courses and has published reference books in the fields of marketing, retailing, and library research. In addition to her daily expcsure to the information needs of students and faculty, she has considerable graphics expertise as Editor of Library Publications. Her extensive teaching experience includes courses in business research and online information retrieval as well as course-related instruction. .

Madeline Andrea Ford, project associate, holds a B.B.A. in Computer Information Systems, and as a spin-off of this project became a student at Rutgers Graduate School of Library and Information Science. She has recently accepted a HyperCard project consultancy with the Rutgers Library as an independent project.

Appendix

a.Student Assignment

. b. Informational flyer to Library Staff Cooperating

c. Student publicity

# CONSULTANT'S REPORT ON MASS TO CLASS HYPERCARD PROGRAM
# BARUCH COLLEGE, CITY UNIVERSITY OF NEW YORK

Martin Kesselman, Coordinator, Electronic Reference/Instructional Services
Rutgers University Library of Science and Medicine

## Background

This report is based upon discussions and observations made during my visit to Baruch College on August 1, 1990 as well as upon further evaluation of the Mass to Class Hypercard program, student reports, and various other documents related to the project. In this report I discuss the following major areas: an assessment of the Mass to Class program in relation to the goals for the program set out in the Title II B Library Research and Demonstration grant proposal; testing of the program by students including a review of student projects; an examination of the Mass to Class program in terms of use of the Macintosh/Hypercard interface; and suggestions for future directions and needs.

## The Mass To Class Program

From my observations of the program, I am of the opinion that the Baruch College Library is on the leading edge in the provision of new library instructional services. The Baruch College Library is forward looking in viewing the integration of computerized instruction as a way to help teach students to become information literate in today's hi-tech world. The major goal of the project as stated in the original grant proposal was to "show the efficacious use of an associative retrieval model for bibliographic instruction in a computer environment using hypermedia technology." I am happy to report that this goal was accomplished. The project also shows the promise of meeting many of the sub-goals and objectives originally proposed such as reaching greater numbers of students, introducing students to new information technologies, promoting the learning of life-long information skills, promoting the use of hypermedia for bibliographic instruction, integrating hypermedia instructional programs into the library system, and to widely disseminate the results of the Mass to Class project.

Mass to Class provides a multi-faceted problem-centered approach to learning information skills. Through the use of various resources and library tools, the students are asked to identify a potential site for a new book store. The program developed teaches library skills needed for an undergraduate retailing assignment rather than for teaching the use of the library per se. It is the computerized equivalent of course integrated library instruction.

The program builds upon the past successes of Baruch College Library in the use of course integrated instruction by extending the idea of the teaching packages Baruch had developed and distributed to over 300 teaching faculty in the early 1980s. Only in this case the teaching package is the Hypercard program. The program also builds upon Baruch's past successes in obtaining grants and other monies to provide innovative automation projects such as those which were used to develop an online classroom and an electronic library research facility for graduate Business students. The program fits in well with the expanding use of electronic information sources by the library such as the OPAC, CD-ROM databases, Dow Jones News Retrieval and other online services.

As was noted in the grant proposal, nearly 20,000 Baruch students and faculty are not reached by traditional library instruction methods. Computer assisted instruction programs like Mass to Class can be available whenever the library is open and used by student at their own pace and at times convenient for them. For this reason alone, the Mass to Class project is important as a demonstration prototype. As a demonstration project, the budget was quite sound. In my opinion, the project team accomplished a great deal with a minimal budget. For the developers this was very much a learning experience which has been extremely valuable and has spawned the development of other Hypercard programs, and has the potential to lead to several other worthwhile library instructional projects in the future.

Instructional Factors

From a review of the subject content provided within the program, Mass to Class, in my opinion, meets proficiency in library research at the functional level (level 2) on the Baruch Library Research Proficiency Scale. At this level, with the use of the program, students can identify relevant material, use library materials effectively to analyze a topic and be somewhat self-sufficient in the use of the library. This is my opinion from reviewing the program and student reports. Unfortunately because of problems with evaluation design and bias, no significant quantitative evaluation of student use of the Mass to Class program could be made at this time.

Ideally, a quantitative evaluation of the project should be based on the use of a controlled study of three groups: one that uses the Mass to Class Program, one that has a traditional library instructional lecture, and one, a control group, that receives no instruction. Pretests and posttests are necessary to determine measurable starting points as well as improvements based on learning. One setback that may have prevented further testing this past semester was that one of the teaching faculty responsible for the case study sustained an injury that prevented him from being available for much of the project.

Sample size is another factor that prevents any significant statements about the effectiveness of the program. 34 students is a very small sample size and then breaking these students down further to 11 groups makes the sample too small for anything but superficial inferences. A comparison of the groups is basically meaningless without a pretest/posttest, as their project results were in many ways constrained. Whereas in an actual simulation, students would have several alternatives to consider, there was only one option for this project and all the available materials included only this option.

Another reason it is difficult to say anything meaningful about the results from using the program or from attendance at the library instructional class is due to the many biases that existed. Several of these biases are noted below:

•The role of prior experience in the use of the library or attendance at prior library instructional sessions was unknown. It was noted by the project team that approximately 50% to 60% of the students had previous library experience either from another business class or from Freshman English composition classes. An introductory business class may be better to test the effectiveness of a program like this in the future.

•Because one class was divided into two rather than the use of separate classes as was originally intended, there may have been biases due to discussions between groups within the class.

•In both groups many of the tasks were divided so that some students were responsible for parts of the library research and others were involved in writing up the report.

•Materials were provided to both groups in the form of a vertical file which included most of the background information on Cincinnati (maps, brochures, newspaper clippings, etc.) and online searches (e.g. on book industry in ABI/Inform, and on Cincinnati from Donnelley Demographics, Cendata and the Electronic Yellow Pages). As this information was available to both groups directly, it is extremely difficult to find out if they learned about these sources through the Hypercard program, in class .or just by browsing through the vertical file.

•Students were given a great deal of information apart from the Hypercard program or class, especially with the Mass to Class Bibliography of Library Resources which was available to both groups.

•There may have been additional bias from teaching faculty who may have provided too much information in selecting what to show students in class.

The groups that went through Mass to Class (alpha group), spent an average of 1 hr. and 45 mins., about the same length as the in-class library workshop beta groups attended. The content of the Mass to Class program was roughly equivalent to the library instructional class which utilized transparencies of various library research tools. In the groups that had classroom instruction, all individuals received equivalent instruction. However, with the Hypercard groups, this is basically unknown as some individuals probably used the program carefully, and others did not. As many groups divided responsibilities, it is also difficult to determine how much time each individual in the groups spent in using the program and specifically which parts of the program were used, which were most useful, least useful, etc.

From an examination of group reports, I was unable to discern any significant differences in quality between the two groups. As often happens in pilot projects, a formal evaluation was not possible. Instead, the comments from the groups about content and effectiveness as well as my appraisal of the program's presentation and use of Hypercard features, discussed in the section below, will have to suffice.

Some qualitative comments about the effectiveness of the Mass to Class program as well as of the library instructional class were listed at the end of the student group reports. It appears that alpha and beta groups spent about the same amount of time in the library working on the assignment (alpha groups averaged 8 hours and the beta groups averaged 7 hours). Ratings were higher for the effectiveness of the library workshop by the beta group with an average score of 1.8 (very important) in comparison to the average rating of the Hypercard program by the alpha group (3.5 or in the middle between important and insignificant). Both groups gave very high marks to the bibliography handed out (all of the alpha group gave this a 1.0 and the average for the beta group was 1.3).

Additional comments were obtained from the alpha group from a pizza party/debriefing session organized by the project team. Some students had false expectations about the role of the Hypercard program and expected the computer to have the actual answers for their projects rather than just explanations of various library tools. There was much impatience expressed in needing to go through the Mass to Class program to locate those library tools that would be most useful. Several students voiced the fact that the program would be more important to use had they needed to work alone rather than in groups. These students also felt that the program, by providing various options and pathways, could in some cases be equivalent to group problem solving. Students in the debriefing session also felt too much explanation was provided in the program and that an index feature would have been useful.

It is interesting to note that several alpha and beta groups liked the way the program or the class presented various library resources but found that the bibliography was most helpful by reviewing the many available library resources that might be useful for the project. In fact, both groups were so enthusiastic with the bibliography that they much preferred it over either the Hypercard program or the library instructional class. Most successful use of the Hypercard program was expressed by those students that viewed the program as a general orientation and who then used the bibliography to aid them in their actual library research.

Hypercard Features

The major advantage of using Hypercard is that it lends itself to browsability. Students do not necessarily need to go through the whole program and instead take from it what they need. In developing Hypercard programs, it is important to determine how a user will need and want to move through the information provided. This depends on various preferred outcomes, such as the learning of particular concepts, the need to locate particular facts, or a topic that encourages browsing and exploration. Another major advantage of Hypercard is that there are many options for updating, revision, and expansion.

Mass to Class is organized into 18 stacks and utilizes a great many of Hypertalk's bells and whistles. There is a great deal of information provided by the program. There are also many features I liked in the program, notably the online simulation, the tutorial on the ABI/Inform CD-ROM database and the glossary (which should be available throughout the program). However, there are many problems with navigation, orientation, and presentation within the program that need to be addressed and corrected. Some of the general comments and specific examples noted below are based on suggestions by Linda Hartley, a temporary librarian at the Library of Science and Medicine, who is in the process of learning Hypercard and the Hypertalk scripting language.

Ideally, the structure of a Hypercard program should be apparent up front whether it be linear, tree-structured with branching, non-linear with no predictable order (which I don't believe is useful for most library instructional purposes), or a combination of these types. Because of time constraints, no preliminary observations of users could be made by the project team before the completed program was needed for the class. Ideally, the producers would observe how first time users navigate through the system and any difficulties encountered.

Besides the brief introduction to the Macintosh the program also needs an overview as to how it is arranged. This could be developed with a slide show capability so that first-time users get a sense of the program even before they learn to use the Macintosh and mouse. At this point the objectives of the program should also be clearly stated. It should be very clear to the user, from the very beginning, how to get around, how to locate what they need in the stack and then how to get out. Possibly using various table of contents cards (sub-menus) just below the home card could help with this. More detail on this point is provided below.

The home card is poorly done. This is a major shortcoming as this card is the heart of the program and should give a clear overview of the stack. The home card is visually dark and dreary and menu choices are unclear. At first glance, the home card looks as though it's a menu for How to Use Databases. Too much space is wasted on the picture of books and not enough space is devoted to setting off entries from one another.

In many cases there is poor maneuverability around the stack. Ideally there should be branching so that users can return to the last menu of the hierarchy and from there, back to home. Instead, the entries on the home card plunge one right into the subject matter cards which must be read one after the other. There are few opportunities to return to a place within the stack outside of home. To remedy this situation, more escapes (e.g. to major cards in each stack) and more menus of choices are necessary. Otherwise users can easily become lost and not know where they are in the stack. Additionally, students should be able to bypass screens not of interest quickly. Here are some examples:

- A very well done card is the one on Business Indexes (Card ID#5611). The title is big and bold and menu options are clearly listed.
- With the Demographics stack, the first card should be like Card ID#6014 which offers options for branching to statistical sources and newspaper and journal articles. This card should also include an option for background information on what demographics is all about, instead of requiring students to plow through all the demographics cards before reaching this point.
- A similar case is within the Statistics stack. There needs to be a card right at the beginning of the stack with a menu which includes links to cards on Federal statistics, state statistics, private sector, and census. As an aside, it's surprising the program doesn't provide a link to information on the Cendata databases from any of the cards on census.
- The cards dealing with smaller geographic areas in the census section, should have a main menu card with a listing of types of sources -- zip codes, SMSAs, FIPs, etc. allowing users to just look at those cards of interest rather than having to click through numerous cards of specific sources.

Better frames of reference are also needed. The logical arrangement of the stacks are not always apparent. It's important for users to know where they are, where they can go, and where they've already been. One help may be to use a common title and even page numbers on all the cards dealing with a particular concept as this helps focus the information for the user and keeps them from feeling lost. Bookmarks in the stack could also be provided as a frame of reference and so that a user can walk away and come back to the program at a later time at the point where they left off.

There is much inconsistency in the format of cards used by the program which can also be very confusing to users. All cards dealing with a similar physical format (e.g. books, databases, etc.) should have a similar physical format (e.g. like cards in the card catalog). Related topics (e.g. discussions of various library resources) should follow the same basic format including placement of fields, button icons, use of graphics, fonts, etc. Instead, there is little consistency in the placement and use of fields and buttons which makes it confusing to users. One example is Card ID#2939 (Statistical Data and How to Find It) and Card ID#7115 (Basic to Statistics is the U.S. Census). Both of these cards are similar in their physical layout however they are not equal in the hierarchy and do not work the same. The second card is a subtopic of the first. While the first card has the user click on an arrow, the second card has the user click on a graphic.

Some cards provide too much detail. Instead, I suggest you provide a button for those users that wish more detail instead of requiring users to have to do a lot of reading of long scrolling fields. I think a good rule to follow is not to have more than one idea per card. Crowding of text with too much information exists in several cards. Scrolling fields should be used at a minimum and then only for those users who might wish more detailed information. Elaborate graphics, including too many buttons and options, also need to be avoided as they can very often be distracting and misleading to users rather than helping them. Two examples follow:

•A case in point is the very confusing Card ID#5207 (D&B Donelley Demographics), the first card in the Database stack. This is also one of the first subject cards most users are likely to choose, as Databases is the second entry on the home card after How to Use. In this card, students are presented with a description of the Donnelley Demographics database with numerous icon buttons. Instead, users should be taken to an introductory screen on databases. To make it less confusing there should be separate sub-menu cards that include a directory of databases (ABI/Inform on CD-ROM should be included here too), one for the online demo, and one for Boolean operators. The more databases icon on this screen and the find icon (magnifying glass) used with the sample search are unnecessary.

•A second example is Card ID#5227 (A Handful of Statistical Bets), which might be confusing to some students, as it includes several different resources and multiple formats. The card also would read more clearly if it went top to bottom and if the information was divided into more cards. I also had difficulty in returning to this card when I selected the icon for Donnelley Demographics.

Some cards make poor use of font sizes and spacing. A case in point is Card ID#2232 (Site Selection). This card is difficult to read and instead of serving as a reminder for users to remember indexes, databases, and CD-ROM, a user is more likely to be confused with the line remember indexes (i.e. what are remember indexes?). Hypercard 2.0 may help somewhat as it will add many useful features such as the ability to have different fonts and styles in the same field (useful for highlighting), and more true hypertext facilities with the ability for users to click lines, works and text as buttons.

Three important points noted by one of the members of the project team in her presentation at this past year's Macademia conference are at the crux of some needed improvements for feedback for the Mass to Class stacks: interactive testing, a notebook feature and check points. The need for checkpoints (or bookmarks) has already been noted above. A notebook feature would be invaluable, especially one that could be saved on the user's personal disk for later printing or review. Reviews and summaries of important points would be helpful, either through summary cards or via interactive tests, which are used quite effectively in the CUNY+ Hypercard program.

## Future Directions and Needs

Several programs developed for Mass to Class, such as ABI/Inform and others have been made available within a Library Help Folder for general use within the Baruch College Library. Baruch would like to eventually develop a body of Hypercard materials. There are numerous opportunities to transfer many of the ideas from Mass to Class to other business topics such as management, marketing and career information. A program on career resources has the potential to reach upwards of 1000 students per year. Notably, the CUNY+ program that was recently completed, builds upon the initial successes of Mass to Class and goes several steps further. Graphics and design are utilized very effectively with this program ar.d feedback, with tests, helps reinforce important concepts. The only complaint I had with this program was that the keyboard section was too long without providing an opportunity for users to escape back to the main program screen.

It definitely appears that the Mass to Class program got people's attention. An article on Mass to Class will be featured in a forthcoming Baruch Grants Bulletin and presentations were made at this past year's LACUNY Institute as well as at the Macademia Conference in Rochester, hosted by Apple Computers. The project director is also developing a manual of hints and skills learned, such as those concerned with design elements and graphics, which should prove invaluable to other developers. Wide exposure and sharing of Hypercard programs by libraries is needed. Tapping into HLIP, the National Hypercard Library Instruction Project, is a great step in this direction taken by the project team. I also urge the project team to consider submitting a paper on their unique approach to library instruction to the journal, Research Strategies. With some reworking, the Mass to Class program may have some promise has a commercial product in providing assistance within the retailing industry.

The future outlook is extremely bright with the opening of a new Baruch library where Macintoshes with Hypercard could be used as information kiosks as well as for instructional workstations. This will be essential as no increases in staff are envisioned at this time. The Mass to Class project has paved the way for new Macintosh-based hypermedia projects, an amazing feat, as Baruch College was an IBM shop and there was originally much resistance at Baruch to the Macintosh. With additional monies and equipment, the new Baruch Library could be a showcase for demonstrating the promise hypermedia technology has for library instruction and information. With the addition of laser disk technology, Baruch would have the capabilities to develop actual interactive simulation programs. Based on my assessment of their dedication with the Mass to Class program and its future potential, I'm fully confident that the Baruch College Library will continue to play a very central role in meeting the information and instructional needs of Baruch College students.

The contents of this report were developed under a grant from the Department of Education. However, those contents do not necessarily represent the policy of the Department of Education and you should not assume endorsement by the Federal Government.

# CREATING HYPERCARD PROGRAMS FOR LIBRARY INSTRUCTION

# A BEGINNERS MANUAL

by
Madeline Andrea Ford

Baruch College
City University of New York
1990

## Table of Contents

The contents of this manual were developed under a grant from the Department of Education. However, those contents do not necessarily represent the policy of the Department of Education and you should not assume endorsement by the Federal Government.

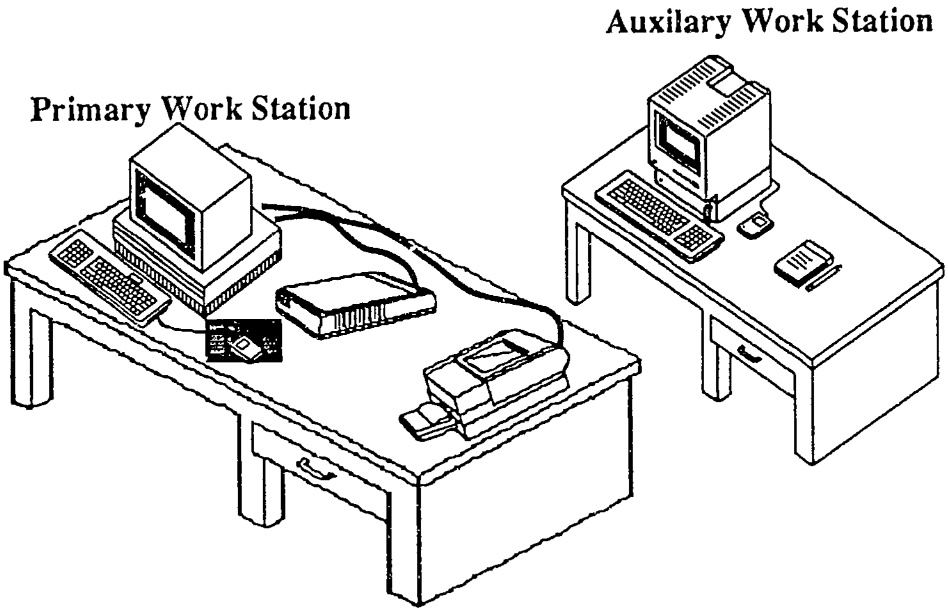
## Introduction.

It should be kept in mind in reading this manual that it describes hardware and software in terms of its usefulness in a particular setting: interactive instruction in effective library research. Obviously both the hardware and software described in this manual have far wider applications. But here we talk of word processors as producing needed text for HyperCard, not for student papers, HyperCard stacks as produced by librarians, not by students.

HyperCard is a database template that comes with a selection of materials and tools to be used as needed. Basic are the "stacks," or files, and the "cards," or records. Any card can be linked to any other card in its stack, or to any other stack. Thus it is a "relational database" template. The producer supplies the data and the links as needed to create the database.

As a medium of instruction the HyperCard program is especially useful for individualized learning since it is self-paced, and can be operated in linear fashion and by branching. Since the branches can be either enriching or remedial, the same program can be used by students with a variety of needs. In a library setting, which traditionally teaches one-on-one, it it more like personal instruction because of this branching capability. Moreover, since library materials tend to have frequent detail changes, its ease of updating is invaluable. Unlike print sources, outdated material is no longer available to the unwitting student.

## I. Hardware
### Project Station



In most production environments, preparing HyperCard stacks will be only one of several uses of the work station. The hardware suggested in this manual in no way portrays the ideal production station, but, based on our experience, offers some suggestions. The production station should be designed to meet the needs of the organization and the user(s). Keep in mind the cost involved in purchasing these components and the strength of each component. A plan should be drawn up to decide what hardware is necessary for successfully developing HyperCard Stacks.

The hardware used for this project reflects what the organization believed to be an efficient yet cost-effective work station made of components available in September 1989 and available at educational prices. Each component has its merits and shortcoming depending on how each is used. The list below gives you the project station designed for "Mass to Class," our retail marketing information sources program:

- •Macintosh IIx (40 megabyte hard drive)
- •Apple Color High-Resolution RGB Monitor
- •Macintosh SE (20 megabyte hard drive)
- •Apple Modem

•Apple Scanner
•LaserWriter IINTX

The power and speed of the MacIIx allows you to move through stacks relatively fast, making it easier to review material effectively. The hard drive is large enough to hold many application programs letting the user have easy accessibility to graphics, fonts, and scanned images without disk swapping. The screen size lends to easier production of projects, while the combination of speed and good visual impact reduces worker fatigue.

The Apple Color High-Resolution RGB Monitor has many virtues even in the monochrome world of HyperCard 1.2.2. The next version of HyperCard does however, have color capability and ability to work with that version was built into our design consideration.

When purchasing the Macintosh IIx, be aware that only the MacIIx system and a mouse are in the bundle. The purchaser should allocate enough funds for the ıonitor and the keyboard.

The Macintosh SE is slower than the IIx and thus is more fatiguing to design on. But because it was the machine most commonly available to students, the HyperCard programs produced at the MacIIx work station had to work on the smaller machine. The SE was used to test and to allow two people to work at the station simultaneously.

The scanner is beneficial to the project work station. Images not available from graphic software can be scanned to create a digitized image. A scanner enhances the overall usefulness of the work station.

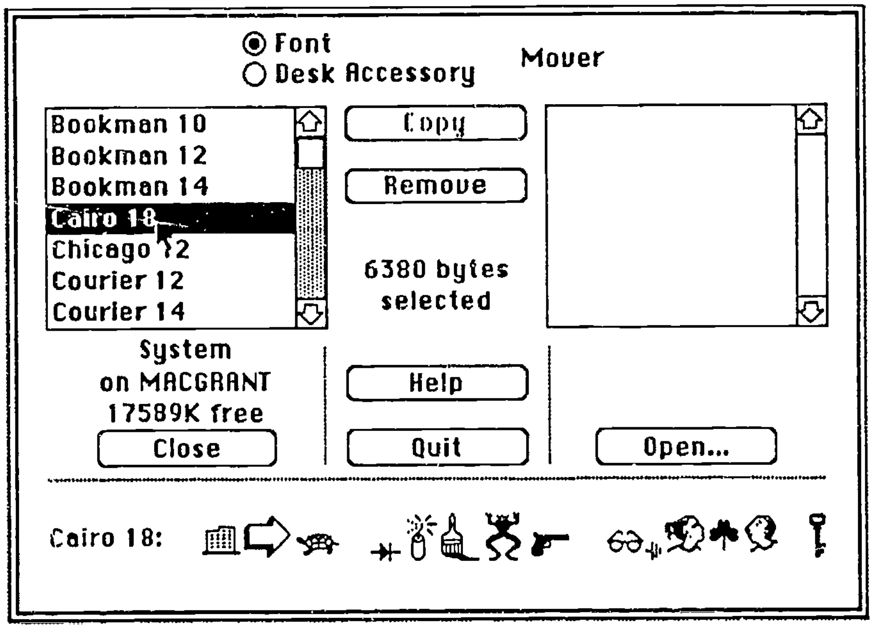
The modem was connected to a Macintosh work station to permit access to commercial databases to electronic mail and to the university's online library catalog.

The LaserWriter IINTX allowed production of various print products in connection with the HyperCard program. Flyers, announcements and "drafts" of the HyperCard program were printed on this high quality printer. The best available printer should be a part of the production station.

The purchase of a dot-matrix printer may be beneficial for student use. The dot-matrix printer acts as a draft copy so that students can print out notes taken while working with a HyperCard program.

## Fonts for the System

HyperCard uses fonts in the system. Thes must be installed in or removed from the system through the Font/DA (Desk Accessory) Mover. The instructions for using the Font/DA Mover are found in the *Macintosh Utilities User's Guide*. Once the DA Mover has been loaded navigating is fairly simple. The Font/DA Mover will list those fonts that are on the system. Help is available through the dialog box. The diagram below represents the dialog box for the Font/DA Mover.



● Font
○ Desk Accessory

Bookman 10
Bookman 12
Bookman 14
Cairo 18
Chicago 12
Courier 12
Courier 14

6380 bytes selected

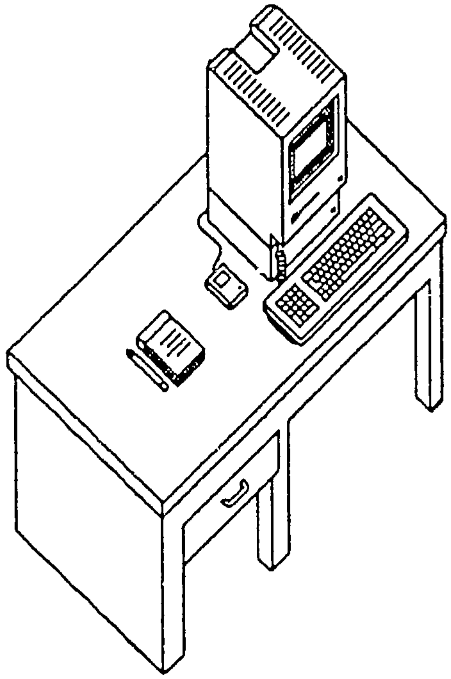
System
on MACGRANT
17589K free

Cairo 18:

Font - *Cairo*, Size - 18, Number of bytes - 6380

The box on the left displays the fonts the system contains. The box on the right would contain the fonts that are to be transferred to the system. When a font is chosen the dialog box provides a sample text in that font style and size, and the number of bytes the font utilizes. The example above displays the font Cairo and all of its characteristics. From here you can either copy or remove the font of you choice. If more than one size of the font is to be copied this can be done by holding down the shift key and simultaneously clicking and dragging the mouse on the fonts that are wanted. Further discussion about using fonts in HyperCard stacks can be found in Chapter 4.

# Student Work Station



The design of the student work station should be simple so that students can work independently. The student station should only contain those components which are necessary for the HyperCard stacks to function. For the "Mass to Class" project the student was provided with these components.

- •Macintosh SE
- •Mouse

The keyboard was removed from workstations in the library, and set at right angles to the Macintosh Lab machines, to emphasize the all-mouse use of these HyperCard programs. Our users were new to using the mouse and we were afraid that some students would waste their time trying to use the more familiar keyboard and thus losing the power of "point and click."

Because the mouse is the only navigational tool a user needs to explore the HyperCard stack, the user can focus exploration and learning on the "card" on the screen. The developer designs stacks with this in mind–all options must show on the "card."

The Macintosh SE with a hard drive is recommended so that the application program can be stored on the hard disk and the stack either on the hard drive or on floppies. Having the application on the hard disk eliminates disk handling by the student. The Macintosh SE is ideal for student work stations because of the compact size of the computer allows the creation of multiple workstations in little space

The work station for "Mass to Class" consisted of the Macintosh SE and the mouse. This allowed for less exploring outside of the stack by users and more concentration on the project. The student benefited because only one navigational tool was used throughout the program and no keyborad commands had to be learned. The designer, however, must incorporate all needed commands into card and stack "buttons" to be activated by the mouse.

The keyboard remained attached for those persons who were setting and exiting the system. Keep in mind that the same functions can be carried out by using the mouse and adapting the stacks to include such a function. To do this the developer is required to create a button which would exit the program or create a function which would restart the program after being idle for a period of time.

## II.Software

Software represents a major investment. Careful selection, using reviews and word-of-mouth recommendations can keep the software to a hardworking few. We found that expensive gadgets exist in the world of software too. Categories of software to consider are:

A. Word Processing
B. Graphic Art
C. HyperCard Related Enhancers
D. Art Tools
E. Telecommunications

The software which was used in the production of "Mass to Class" are divided into categories below.

### A. Word Processing

•MacWrite II
•WordPerfect

### B. Graphic Art

•Click Art Business Images
•Images with Impact

### C. Art Tools

•MacPaint
•MacDraw II

### D. HyperCard Related Enhancers

- 101 Scripts and Buttons
- Stack Magic
- HyperAtlas
- HyperAnimator
- HyperCard Stacks

E. **Telecommunications**
Hayes SmartCom II

## A. Word Processor

A general use word processor is useful. Here MacWrite II was chosen for a near zero learning curve while WordPerfect was bought because it is the "house" program. In producing a teaching stack, it is often useful to input text where it can be run through the spelling checker; on occasion the text needs to be read by an expert and revised. These tasks are meat and drink to the word processor. The text can then be imported into a HyperCard stack. For short excerpts all that is necessary is to copy (Command C) after highlighting, exit and load HyperCard. Create a field to receive the text, click in the field and paste (Command V) the text into the field. For several texts, copy into the Apple Scrapbook first then follow the same procedure. For long texts create a scrolling field and then follow the same procedure for pasting the text.

WordPerfect in comparison to MacWrite II is about the same except there are enhanced features provided for word processing. WordPerfect can be command driven from the keyboard or mouse driven. The learning curve is slightly higher.

During your word processing you may decide to use various fonts for more effect, if so the text will default to the set font for HyperCard. To circumvent this use the default font for wordprocessing. When creating the field choose the font that will be appropriate.

## B. Graphics

Graphics are very important to stack development. There are various graphics available to relate ideas. The two chosen for this project were directed toward business. Images in these packages can be modified for use in the stacks and allow use without violating copyright. Read their documentation. Graphic software can be accessed independently or through another application. Verify that you have the necessary support application. The verification will save time and money if the developer knows the additional software needed to use the graphic art. Various types include PICT, Uncompressed TIFF, Compressed TIFF, and MacPaint.

Click Art ® Business Images can be accessed by two methods. The first is through HyperCard where a menu is provided to locate the graphic . The copy/paste method remains the same. The ability to remain within HyperCard makes this graphic application a snap to use. The other method is via the MacPaint application. When tried the individual MacPaint files could be accessed without having the application present. This allows for accessibility without the purchase of MacPaint. One pleasing feature of Click Art ® Business Images is the DA Click Art ® Clip Out. This feature allows the developer to continue working in a stack or application without having to open another stack for graphics. The developer can open any existing MacPaint files for use. The Clip Out DA allows for minor adjustments to the graphic. Some features are erasing, penciling in lines, cutting sections out of the graphic, making the window smaller to see the entire contents of the graphic, making multiple copies of a graphic by holding down the option and command key together and dragging the image across the screen, making a duplicate copy of the image by holding down the option key.

Images with Impact must be accessed through an application program which supports EPS (Encapsulated Postscript). The list comprises of those applications which supports EPS.

Drawing

Adobe Illustrator 88 ™
Aldus Freehand ™ (Ver. 2.0)
Cricket Draw (cannot edit and separate graphics)

Page Layout

Aldus PageMaker ®
Letraset Ready, Set, Go! ®
QuarkXPress ™

Presentation

Aldus Persuasion ™
Cricket Presents
Letraset StandOut!™
Microsoft® PowerPoint ™

The list above provides additional software that must be purchased in order to take advantage of the graphics in Images with Impact. The method of accessing the graphics depends entirely on the EPS supported software instructions.

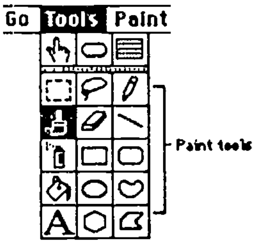
## C. Art Tools

The two applications listed above allows, the developer to create various images. These two applications are very helpful when designing floor plans or any type of elaborate drawing. The graphics created can be easily placed into stacks.

MacDraw II is a very good application program. Text and graphics can be integrated together easily. A spell checker is also provided lending to corrections of text. Each graphic is treated individually making movement much more flexible. MacDraw II at times can be constraining, because, there is limited versatility in the handling of shapes and pattern use.

MacPaint is also a good application program allowing the developer to perform similar functions as MacDraw II. MacPaint allows the developer to create free forming shapes in various textures. MacPaint does not provide a spell checker because the text is treated as an image.

HyperCard can also be used as an art tool. Under the menu tools there is a palette which looks like this:



The top three tools are those which control the browse, button and field modes. The remaining are the selection tools the lasso and selection box and art tools. A developer can create shapes, design custom background and add patterns to the work. This feature is similar to how MacPaint operates. One constraint is that any given image cannot be larger than the card. For the developer who wants the drawing to be precise, a special feature exist FatBits which enlarges the image to show each bit on the screen. Correcting or enhancing the image without causing disorder to the rest of the image is thus made easy.

## D. HyperCard

The flexibility available in HyperCard can prove very beneficial. The ability to open various stacks within the same application makes movement simple. Many types of software are available ranging from those that concentrate on buttons to those that have developed scripts to perform actions. Stacks have been designed so that they can be customized to meet the needs of the developer's project.

101 Scripts and Buttons provides the developer with various buttons, cursors and scripts to be used together or individually. The buttons can perform various operations and can be applied to any stack. Some buttons perform an action to give the user the impression something is happening. Cursors can be chosen in order to demonstrate a particular idea in a stack or depict a special use. Cursors can be helpful when the developer wants to demonstrate to the user a button; the cursor can change from one design to another. The scripts are very helpful because they have already been coded to perform an explicit action. These scripts can perform various roles such as allowing specific words to be boldface within the same field.

One arduous task in using 101 is making sure that all the attributes to buttons or scripts have been copied accurately. For some of the buttons and scripts this cannot be done because of internal controls. The instructions on how to do this are limited. The developer must use good judgment and trial and error.

Stack Magic is a very handy collection. It is not available on the market but through the publication *MacWorld* who developed the stack. The stack has five areas: Buttons, Cursors, Digitized Sounds, and Utilities. The button available range from simplistic to highly animated. The cursors available also come in a wide variety. There are cursors which will relate a specific idea or several ideas. The digitized sound is good. This allows for more interaction between end user and computer when responses can be given. The sound can enhance a stack when not used excessively. The utilities are helpful to the developer who can perform various functions effortlessly. One function is to split a stack, which allows the developer to extract any number of cards to form a new stack.

The copying of these functions is very simple. The stack has been designed so that a script will perform the task automatically when chosen.

HyperAtlas is designed for those developers concentrating on geography. The stack includes the maps for the United States and for the World. The stack has been developed so that it can stand alone or within an existing stack. Minor changes must be made in order to incorporate the cards into a developing stack. Cards have been designed with formats already laid out, only the data has to be keyed in the fields. The software is fairly simple to modify with the support of the instructions.

HyperAnimator is an animation application. Images and sounds are used together to bring life to characters. The application is fairly simple to use. The instructions in the user manual are straight-forward and easy to apply to the stack. HyperAnimator allows the developer to create characters or modify existing characters. One failure of HyperAnimator is the inability to have characters conversing together. One character must finish before another can begin. The sound must be modified ~o that the character will not pronounce each letter, but pronounce the word. This is done through the speech synthesizer which creates the digitized sound. HyperAnimator is a excellent addition to a software purchase. HyperAnimator provides more interaction and enhances any developing stack. I have created a simple step by step guide to creating a custom stack for HyperAnimator. The manual, of course, gives a more in depth explanation. This is not recommended as the first effect you make, but is given as an example of how powerful the scripting program is.

How to create an animated character:

1. Create a new stack to save the actor in
2. Double click on HyperAnimator
3. Click on Casting Call
4 . Choose "Place Actor in Dressing Room"
5. Open File Curtain Call and choose an actor
6. Make sure when actor is in dressing room the actor has a mouth at the rest position.
7. If there is no mouth then choose one from the one of the positions on the right
8. Copy the mouth, go back to the rest position by clicking on the rest button, paste the mouth and choose browse tool.
9. Click on the stage button
10. Check to make sure that all the features and mouth movements work by typing in sample text
11. Click on the spotlight button
12. Name the actor and click on the save actor button
13. Select the stack you created earlier to save the actor in
14. Quit HyperAnimator
15. Open the stack you created
16. Create a new button and name it after the actor -- this button will show the the actor on the screen
17. Type in or copy the following,filling in the name of the actor where *name* occurs.

    RAVE "{ACTOR *name* AT LEFT 16 TOP 16}" --these are the coordinates to place actor at)

    RAVE "{SHOW *name*}"
18. Create another button and name it talk.-- This is the button which activates the actor to speak

19. Type in
RAVE "{RAVE *card field* "text"}" --the text can be changed to say whatever you want
RAVE "{RETIRE *name*}"
20. Create field and give it the same name as the ta!k button
21. Type text into the field for the actor to recite.

At the right moment, a user will click on the button and have an animated presentation of what he has to know.

HyperCard stacks are included with the HyperCard application. The most frequently used stacks for "Mass to Class" were art, button, card and stack ideas. There is quick accessibility to the stacks and copying is simple. The developer can locate many useful graphics, scripts and layouts for a developing HyperCard stack.

These graphics can focus attention or become buttons to do something, show something or move to another card or stack.

The software listed describe what was used in "Mass to Class." Other software applications may meet the needs of the developer and should be considered. Keep in mind the cost of the software and if the software can be used independently.

## E. Telecommunications

SmartComII is a telecommunications software for the Apple Macintosh which allows access to commercial information services, electronic mail, mainframes and other microcomputers. The software provides a manual which directs the developer on the setup of the application. Setting up the application is done through a series of questions found in the dialog box.

For the "Mass to Class" project SmartComII was used for online searching in several databases, reading mail and downloading screens from the University Online Catalog. The most beneficial aspect of SmartCom was the ease of use in downloading the screens of information.

### Downloading Data from SrıartComII

There are two procedures for downlcading. The first is to download everything described by the type commana; e.g. t5/5/1-5. The second is to download highlighted areas as displayed on the screen.

Procedure One:

The first, procedure simply requires clicking on the icon marked with a disk and paper. 💾 Name the file and click on the button marked "capture", this will download all the data into an ASCII file which can then be transferred to a wordprocessing application for editing.

Procedure Two:
1. Begin by highlighting the text that you want downloaded. To do this move the cursor to the position to be highlighted. Click and hold the mouse, then drag the mouse to the end position of the text. When you release the mouse, the data should be highlighted. If some of the data is missing you can continue dragging by holding the shift key and positioning the mouse on additional data to be highlighted.
2. Click on the icon with the paper and the disk, name the file and click on the capture button.
3. To add additional data to the file, repeat step 1-2. Another dialog box will appear asking you to either add-on, replace, or cancel. Click on the add-on button. This will append the data directly to the last part of data saved in the file.

Using the second procedure can save time by cutting down on editing the text. Only the data needed is downloaded.

### MultiFinder

When there is need to move from one appliation to another and back again, typically you have to exit from one program, load the other, then exit from that one to load the first.. With MultiFinder, a special feature for the Macintosh family, the user can work in various applications simultaneously. The need to exit one application to enter another is avoided.

A word of caution, to use MultiFinder the Macintosh computer must have enough RAM (Random Access Memory) to function properly. Without enough RAM only a certain number of applications can be opened at the same time.

**How to Use MultiFinder:**

Go to main desktop screen (Quit from open applications)
Pull down the menu **Special**
Choose **Set Startup**
There are two choices:
1. **Finder** and 2. **MultiFinder**
Click in the **MultiFinder** box
Close the dialog box
To activate MultiFinder the system must be rebooted
Pull down the menu **Special.**
Choose **Restart.**

This will automatically restart the system with MultiFinder on. You will see at the top right hand corner of the screen a miniature icon computer, this indicates that MultiFinder is on. When you open an application the icon on the top right hand of the screen changes to reflect the type of application you are working in. To choose another application click on the icon until the icon changes to the application you want or go to the miniature computer icon, go back to the main desktop screen and choose another application. To see the applications that are open pull down the Apple Menu (the little apple). This displays the application that are in use. By highlighting the application, it can be opened. For a more detailed explanation of MultiFinder consult the *User's Guide* manual.

# III. Designing the HyperCard Program

## Designing Content

The data which will be in the stack should be laid out. Notes should be made about the size of the text and how additional data will be accessed. Decisions should be made about points to be covered and what kinds of information should be imparted about each. Where similar types of data occur, identical presentations should be used to facilitate learning. It was the original intent of the designers of HyperCard that new content and design would suggest themselves as the stacks were made. Ease in editing allows for improvement as the project develops. In fairness to the learner using the program, however, a final editing for consistency has to be made.

During the design phase a flowchart should be kept. This gives the developer the chance to see where the project is going. The developer has a visual template to see where links are made and how the navigation will flow. The developer can than decide on the types of visuals. Some should remain consistent throughout the stack, while others will be specific to one module. Buttons which will be used throughout the stack should be labeled. These buttons should not be used to convey another message. For instance, the explanatory material for each subject area was linked to a glossary stack, using the same kind of visual clue to consult the glossary.

*Examples:*

| | |
|---|---|
| Going to the menu card | Visual dissolve |
| Going to a new section | Visual checkerboard |
| Right large arrow | Goes to the next card |
| Computer | Goes to the menu card |

These are just a few examples, listing visuals and buttons for the stack helps make programming easier and faster. The developer can anticipate what will happen and the hypertext remains the same throughout the stack so that the hypertext can be copied form card to card.

The ability for the end user to choose a path of operation is essential. This allows for free exploration of the project. The end user is not confined to a linear movement but has optional branching. For the "Mass to Class" project this was accomplished by designing a menu. Although the student could follow a lenear approach by moving steadily through the menu choices, it was also possible to explore any area. The design also considered the student who needed specific information or one time information.

At the end of a section it is a good idea to have some type of test. This gives the user some time to decide if all consideration have been made. The developer must recognize if such a test is needed depending on the type of stack being created. The end user then has the opportunity to continue the work or to redo that section. The end of each section could also have a notepad where students are able to jot down notes. When the students work is completed a printout could be obtained. This encourages the student to read the material carefully and decide what information is worth remembering.

## Designing the Programming

It may seem strange to think of designing scripts, but consistency is the key to successful programming. Criteria for a successful stack should be based on the card backgrounds, the visuals performed, the fonts used, and the types of buttons. For the "Mass to Class" project the end user was provided with a homemenu card which provided several choices for moving through the stack. This card could always be accessed by clicking on the home button, this was consistent throughout the project and provided the user with a quick doorway to the menu card. It was decided that it would be bad design to provide the home button on each card, instead, the home button was provided in the beginning and end of each section. The home button provided some security in using Mass to Class. The menu could always be accessed by clicking on the home button. Titles were also used to show the user the location in the stack.

The animated visual effects of buttons should remain the same. For instance, the home menu button would always have the checkerboard shift. This consistency subliminally prepared the user for a change of subject: going home .

The menu that is designed should contain the essential pathways of the stack. Those pathways should be able to stand alone, meaning that a user returning for another session can locate the last pathway she was working in. The main menu design should be clear and the pathways recognizable. The developer may have to give hints to the location of buttons, for example "Click in the rectangular box" or "Click on a number". These hints can be located in small type in any area that the user will see readily.

# IV Selecting Graphics

Choosing which graphics to use can be the most frustrating task. There are two considerations in choosing graphics

1. Who is the audience? and
2. Is the graphic understood?

The audience is very important, for a graphic should neither be too confusing or mind-freezingly simple. The choice should be made by examining the audience's age and educational level. The graphic should also be able to stand alone without ambiguity. Some symbols will have to be artitrarily defined for the program, but universals should be sought first.

Icons should relate to the topic or idea, an example STOP confers the message of exiting or leaving a particular section of the program. Further, if the user makes a mistake by clicking on the wrong button there should be a button which will return the user to the previous position. This gives the user the ability to make mistakes without damage to learning the material.

The audience determines the graphics. In some instances , the audience does not accept icons easiiy and would prefer to see words. It is a good idea to have an unprompted opinion of a user, preferably one who will be in contact with the program when it is completed. Does the user misinterpret icons? If so, is it possible to find out why, so that if a replacement is selected, it can profit from the knowledge.

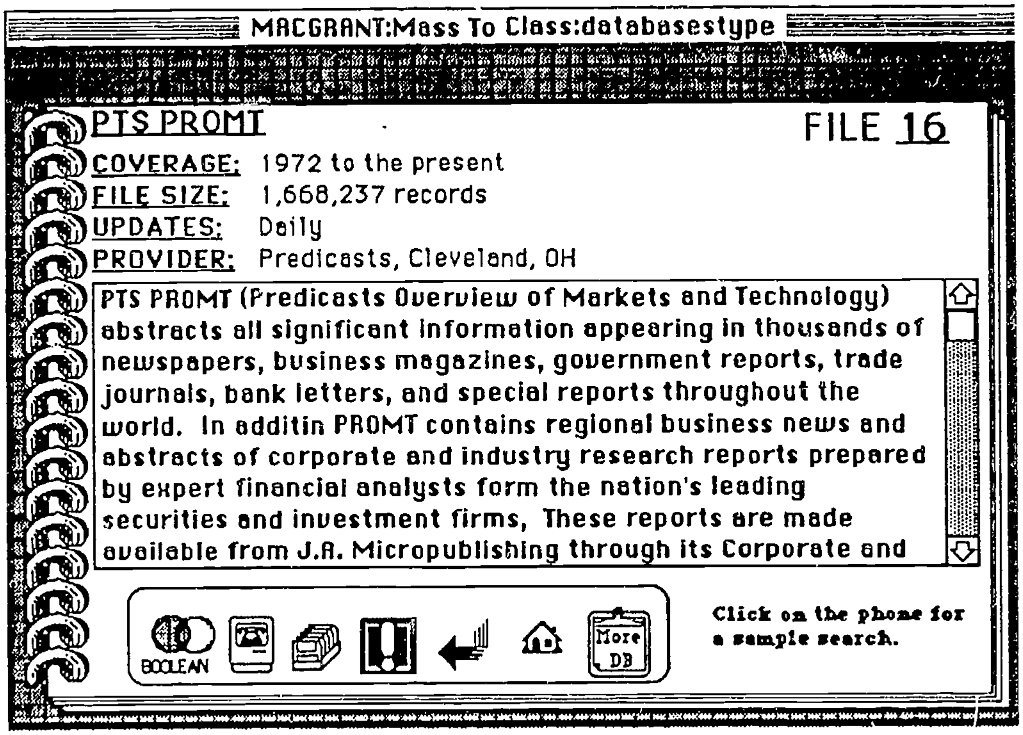
For the "Mass to Class" project graphics were chosen to express specific ideas and portray certain areas wordlessly. Some graphics were specific to the project, using industrial and marketing concepts. While some were chosen from Click Art, other were taken from printed material in the public domain; Dover publishers has many such collections. Those graphics in print format had to be scanned into the computer.

## Scanning Print Sources

When the scanner was purchased it came bundled with two types of software, the first AppleScan and the second HyperScan. AppleScan allows for the scanning of pictures or line drawings. Scanning pictures requires setting the Scan control panel to Halftone or Grayscale. Halftone mixes contrast and brightness with varying degrees of black and white, to create a image similar to the picture. Grayscale uses gray tones to create a image. Line drawings should have the setting of Line Art, this setting uses black lines to define an image. Three fon:nats exist for saving a scanned image PICT, TIFF and MacPaint. To scan follow the directions in the Apple Scanner Owner's Manual.

HyperScan is a scan program designed for HyperCard. Scanning is done by choosing several options. The graphic below provides such an example.

## An Example of Choice of Icons


MACGRANT:Mass To Class:databasestype

PTS PROMT FILE 16

COVERAGE: 1972 to the present

FILE SIZE: 1,668,237 records

UPDATES: Daily

PROVIDER: Predicasts, Cleveland, OH

PTS PROMT (Predicasts Overview of Markets and Technology) abstracts all significant information appearing in thousands of newspapers, business magazines, government reports, trade journals, bank letters, and special reports throughout the world. In additin PROMT contains regional business news and abstracts of corporate and industry research reports prepared by expert financial analysts form the nation's leading securities and investment firms, These reports are made available from J.A. Micropublishing through its Corporate and

Click on the phone for a sample search.

This particular card provides the end user with seven choices. The design was made so that the end user would have to make choices from that card. The student user is in control of the choices which are made with no specific order of operation. No choice is solely dependent on another. Each icon branches to other areas of the application. As can be seen the icons speak for themselves, the only suggestion which is made is about clicking on the telephone in the computer to observe a sample search, since we assume that most undergraduate students do not know about online searching and its use of telephone lines.

## The icon and its scripts.

The icon above directs the end user to information about Boolean logic.

This particular icon was chosen because most of students are familiar with Venn diagrams from their math and/or computer courses.

```
Script of card button id 10 = "Boolean"

on mouseUp
     hide card field dbdescription
     show card field dbdd
     visual iris open to card
     go to card id 11692 of stack "DATABASES"
end mouseUp
```

The script which acts as a mini program referred to as HyperTalk, communicates to HyperCard the actions to execute. This script performs several operations. To perform any script a condition is needed, for this script on mouseUp was used. This indicates to HyperCard that when the user clicks on a button the next step is to perform the script. There are many conditions available such as *on openCard, on mouseDown,* or *on enterKey* each performs a different function depending on the conditions set by the developer. For more on using these conditions consult the user guide *HyperTalk Beginner's Guide: An Introduction to Scripting* or a manual on HyperCard.

The script for this button begins by hiding a card field "dbdescription" then showing another card field "dbdd". Fields can be named or identified by their card field id number which is set automatically by HyperCard. The next line performs a visual effect to help the student focus on the program and act as a guidance tool. Visual effects should be used in moderation and should not be too busy. While the visual is performing the script directs HyperCard to advance to another card "card id 11692" in a different stack "DATABASES". The card is identified by a number, but can be identified by a name. At the end of the script the condition is ended by a "end mouseUp" this indicates to HyperCard that the execution is finished. The user should now be in the stack called "DATABASES."

Scripts can be very simple or very elaborate like the one below, that types words letter by letter to simulate a computer search.

```
Script of card button id 11 = "online"

on mouseUp
     hide card field dbdd
     hide card field "dbdd"
     show card field "number"
     DIAL"679-0332"
     hide card field "number"
     show card field "dbdd"
     visual wipe left slow to card
     go to card id 3061
```

```
    hide card button back
    delete line 1 of card field id 8
    delete line 1 of card field id 12
    delete line 1 of card field id 10
    hide card field "quest1"
    hide card field "quest2"
    hide card field "quest3"
    hide card field "file575"
    hide card field "search"
    wait 20
    show card field "quest1"
    wait 90
    choose browse tool
    get loc of card field id 8
    click at it
    Play Boing Tempo 221 ab7e.
    type "B"
    wait 15
    Play Boing Tempo 221 ab7e.
    type "E"
    wait 15
    Play Boing Tempo 221 ab7e.
    type "G"
    wait 15
    Play Boing Tempo 221 ab7e.
    type "I"
    wait 15
    Play Boing Tempo 221 ab7e.
    type "N"
    wait 15
    .
    .
    .
end mouseUp
```

When the end user chooses the telephone in the computer icon a series of commands will be executed. In the previous example, Boolean, the script ended when HyperCard advanced to another card. This script continues performing various functions. The order of operation is determined by the developer, in the order in which they will be executed. Scripting invokes an underlying program which can be very elaborate. When the script reads DIAL "679-0332," the underlying program provides sounds which vary by number dialed. This level of programming is very efficient. For example, the script above begins by hiding and showing fields and simulating the dialing of a telephone, followed by a visual. The visual goes to a new card and various fields are deleted and hidden. The script continues by typing in the word B-E-G-I-N and the rest of the information. The script is structured to show what each step will be. The developer can read through the script to see the order of the commands; to use parts in another script where appropriate, etc. The script performs a simulated search where the computer acts as a student who is typing in a search and waiting for the response from the database. The end user must simply click on the button and invoke the script to perform the function.

## Fonts for HyperCard Stacks

The fonts that are used in HyperCard stacks reside not in HyperCard but in the system. Each Macintosh station needs the correct fonts installed or a default font will be used and distort the card design. The developer of the stacks should plan out the fonts that will be used in the HyperCard stack. Making this decision from the start will give the user enough time to have the fonts available on the work stations and student stations. Fonts should be used in appealing sizes to reduce eyestrain. The fonts should be clearly visible in the student station. A font in size 9 point is much more difficult to read on a Macintosh SE screen than a size 12 point. The number of fonts used in a stack should be at a minimum. Font selection sho ıld be based on readability and size and availability.

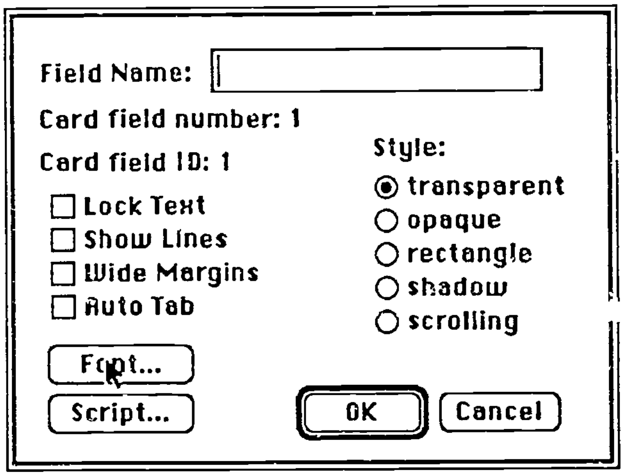
Using fonts in HyperCard stacks is very simple. There are two dialog boxes which give effect to the field and the font chosen. After creating the field, double click in the field, this will activate the first dialog box. There are many choices which can be made at this point. Notice the example below. The field can be named, this is helpful when programming since a field will operate by its number or the name given to the field and a field name is easier to remember when scripting The dialog box also gives the card field number and the card field id number, again this information is helpful for programming. The next set of choices are those under the card field id number.

The first "Lock Text" allows the user to lock the existing text in the field so that the user cannot alter the text in the field. Be sure to do this at the end, after all of the text has been entered or you will have to unlock the field before entering the text. Locking the text is imp ɔrtant because the browse tool (the pointing hand) changes to an I-beam in an unlocked field, which disconcerts the student. The second choice is "Show Lines" which can act as a guide to the developer or enhance screen appearance for the user. The third choice "Wide Margins" increase the space between the lines, providing easier readability, but will reduce the amount of field space for text. The last choice is "Auto Tab" this feature works well when the user or developer has to enter text into the field. When "Auto Tab" is activated text can be entered into the field when the user depresses the return key the cursor jumps to the next available field. The user does not have to reposition the mouse to click in the field and frees the hand for typing in the necessary text. When creating fields be sure to do them in correct sequence, "Auto Tab" will go to each field in the order they were first created. "Auto Tab" will skip locked fields.

The next set of choices sets how the field will look on the screen. The first is "transparent" which puts a plain field on the screen. The second "opaque" which makes the field <u>white</u> so that anything under the field will not show through.

The third "rectangle" creates a rectangular shape for the field, this helps to add dimension to the field and bring attention to the text. The fourth choice is "shadow" which puts a rectangular shape and a shadow around the rectangle. The last choice is "scrolling" which creates a scrolling field for the text. This is a good feature when there is a lot of text with little space on the card to hold the text. The user views part of the text at a time.

The next choice is font this opens a dialog box which gives the developer the opportunity to choose the font, style alignment and size of the font. To change the font click on the button marked font. An example of the dialog. Under that choice is the button "Script", when activated the developer can program the field to do a particular function.



Field Name:

Card field number: 1

Card field ID: 1

□ Lock Text
□ Show Lines
□ Wide Margins
□ Auto Tab

Style:
● transparent
○ opaque
○ rectangle
○ shadow
○ scrolling

This is the dialog box which list the fonts that are available on the system. When the dialog box appears it shows the current status of the field you are in, which in most cases is the default font and size. The developer can make changes to style, align, font and size. A sample is provided to show what the font will look like and how different styles will effect the font. By clicking on your choices all the attributes will be set. You can also, change the size of the font if the size is not available. Sometimes this is not a good choice because the font can be misrepresented. If the font which has been chosen does not meet with approval, simply go back to the dialog box and make changes. The text will be reformatted to show the changes.

For the "Mass to Class' project several fonts were used. The fonts were

New York Times
Chicago Geneva

Among the basic font used by most systems are Geneva, Chicago, New York, Monaco, Helvetica, Courier, and Times. Certain fonts look better on the screen and are pleasing to the eye. It is very difficult for someone to read a font like San

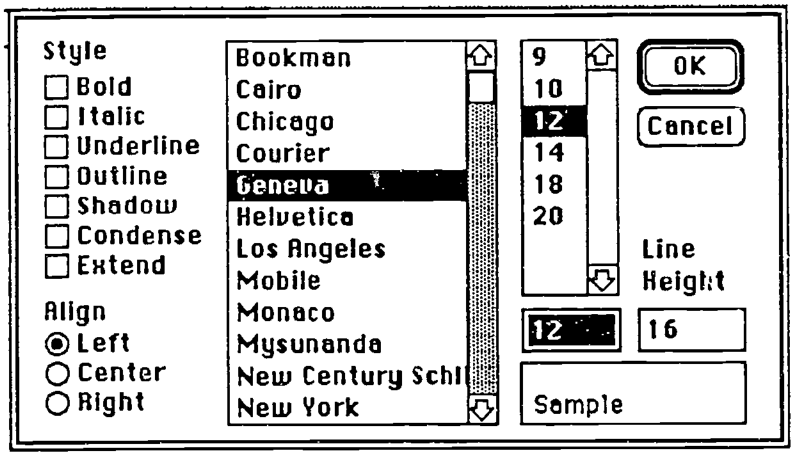
Francisco when it covers the entire screen. However, using the font for a heading gives more impact to the card. The reader moves from the overall topic to a specific expanation.

Use more elaborate fonts, like San Francisco in the text style mode. Text style is a painting tool provided by HyperCard. Any fonts on the system are available through text style. To use text style choose the A from the TOOLS menu. To change the font either drag the tools box from the menu bar and double click on the A or choose from the Edit menu Text style. A dialog box should appear listing all the available fonts to chose from. Choose the font, size, alignment, and style, click on the OK button. Click in a blank area on the card and type in the text. This text can be moved anywhere on the card by lassoing the text.

**Helpful Hint:** If the font chosen is not appropriate go back into the dialog box and choose another font. HyperCard will automatically make the change, without you having to retype the text. This must be done before you move the cursor to another position. If the cursor is moved to another position then the text can only be changed by retyping the text with the new font.

The text style feature is a wonderful tool when used in the appropriate situations. I do not recommend using the text style mode for lengthy pieces of text. The text becomes more difficult to change because the text is treated as a picture. Keeping lengthy text in fields allows for more flexibility, font attributes can be changed by clicking on certain choices. The filed length and width can be moved easily and editing the text does not require erasing the word and aligning the new word with the other text.

These fonts were used in various sizes and styles to give more effect to the card. The number of fonts used should be minimal, more fonts create more confusion. Another major consideration is that each system will not have the same system fonts and those fonts not appearing on the systems will default to the system font



Style
□ Bold
□ Italic
□ Underline
□ Outline
□ Shadow
□ Condense
□ Extend

Align
● Left
○ Center
○ Right

Bookman
Cairo
Chicago
Courier
Geneva
Helvetica
Los Angeles
Mobile
Monaco
Mysunanda
New Century Schl
New York

9
10
12
14
18
20

12

Line Height 16

This is the dialog box for HyperCard which will allows the developer to set the fonts. When a font is chosen all the available sizes for that particular font are listed in the sizes column with the default size highlighted. Changes can be made to the style column. There is no limit to the number of styles which can be chosen. The developer can also set how the text will be arranged in the field from the align column. When all changes have been made simply click on the "okay" button and continue working in the stack. If a larger font size is desired click in the highlighted box and type the desired size for the font.

## Scanning Images

At some point in the development phase, certain images may not be available, difficult to locate in machine readable format, or the software to expensive to purchase. For this reason, the use of the scanner is beneficial. Copyright-free images such as the series that *Dover* publishes can be scanned without breaking copyright laws. The scanner can transfer a hard copy image to a machine readable image. After, the image has been scanned it can be modified to reflect the needs of the developer. Scanned images absorb a tremendous amount of space, this will make stacks very large and can prevent the stack from being saved on a floppy disk. HyperScan will not consume as much space as an image scanned via AppleScan, if possible use HyperScan for HyperCard stacks. There are two methods for scanning images, the first, HyperScan and second, AppleScan.

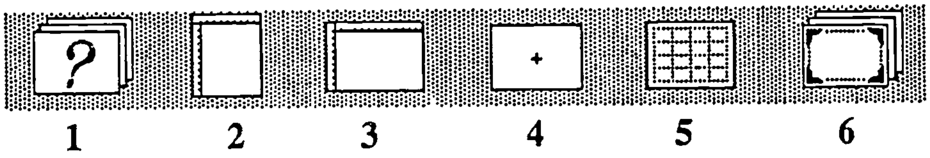
## HyperScan

The HyperScan software is available from the developers of HyperCard. HyperScan enables the developer to scan images directly into HyperCard stacks. Here are some steps to follow to make the scanning process pleasurable.

## Using HyperScan

- Begin by printing the entire Help stack, which will be the manual.
- Read the manual for a basic understanding of doing a scan.
- Go through the tutorial provided by the developers.
- The tutorial takes about 10 minutes, (choose a photograph or some type of line art image to scan) and let's you actually scan the image while going through the tutorial.
- Try scanning an image, refer to the Help button (1) and manual if problems arise.

The HyperCard developers have provided several buttons which make the scanning process faster and easier. The function of each button is provided below.



1. **Help button:** click on this button when a question arises on how to do something.
2. **Portrait:** This indicates to HyperScan that the image will be scanned in a vertical position.
3. **Landscape:** This indicates to HyperScan that the image will be scanned in a horizontal position.
4. **Contrast/Brightness:** Will take you to the last image scanned, and allow the

contrast/brightness to be altered.
5: **Effects:** This button will take you to the twenty-five types of effects the image can have. After making a choice, you are returned to the image with the new effect in place.
6. **Slide show:** This gives a quick slide show of those images which have been scanned by the developers of HyperScan.

**Hints for successful scanning:**

- Black and white photographs, and line art images should be scanned for better results.
- To create the best image move the plus sign to various areas on the card, each degree of movement will change how the image will look. The contrast/brightness mode work together. Refer to the manual for contrast/brightness areas.
- The effects card can be fun or a mass of confusion. The reason being is the amount of choices, twenty-five to be exact.

    Some helpful hints to make the fine tuning process easy:

    For line art (outline or line drawing of an image) use black and white.

    Diffusion is usually the best for pictures of people, it is also the automatic default for HyperScan.

    For a good feel, try using different effects on various images. See which ones meet your needs and are catching to the eye.
- After perfecting the image it is time to save the image find the save button on the screen. There are two ways to save the image.

    **First method** - by simply clicking on the save button, this will append the image to the HyperScan stack. Be careful it is very difficult to delete the image once it has been saved to the stack.

    **Second Method** - the best choice, before clicking on the save button hold down the option key then click on the save button this will activate a message box which states "**This card has been copied. Go where you wish and paste it.**". The image is put into the clipboard and can be added to any stack. There is no need to cut/copy and then paste the image, HyperScan creates a card which can be pasted directly into a stack.

### AppleScan

Read the manual before trying to scan an image. The manual will give step by step instructions for scanning images. Three types of images can be scanned

1. Line Art   2. Halftone   3. Grayscale

Each results in different expression of the image. Below is a list of settings that they can perform, following the list is a explanation of the settings.

1. Line Art
Types of settings: Threshold

2. Halftone
Types of settings: Threshold, Contrast/Brightness, Halftone Pattern, and Graymap

3. Grayscale
Types of settings: Threshold, Contrast/Brightness, and Graymap
(Grayscale blends various gray patterns of the image. This does not print out effectively but looks very good on the screen.)

## Settings

### Threshold:

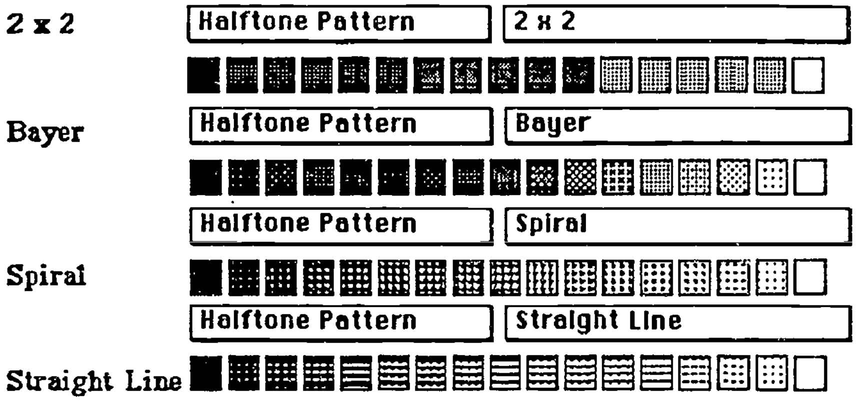
Threshold fills in white space with black dots making the image sharper. The settings range from 1-17, one being the darkest and seventeen being the lightest. For line art the more definition the lines have the better the image will look. Be careful, sometimes making the image sharper can cause blotching where the image has dark spots in some areas. If that happens, change the threshold setting by moving the bar toward the seventeen mark. If you do not do this, the cleanup will have to take place after the image has been completely scanned.

### Contrast/Brightness:

Contrast/Brightness work together. Contrast will make the image sharper or duller, depending on the degree that is chosen. Brightness will make the image darker of lighter. Using the Contrast/Brightness for the first time may prove frustrating, try working with only one at a time, eg. Contrast bar only.

### Halftone Pattern:

Halftone Pattern has four patterns to choose from Spiral, Bayer, Straight Line and 2x2. When a pattern is chosen it will change the image to reflect the new pattern. Examples of each pattern are given below. Try each pattern to see which one provides the best effect for the image. Each pattern will give the image another look which may make the image more expressive.



2 x 2 — Halftone Pattern | 2 x 2

Bayer — Halftone Pattern | Bayer

Spiral — Halftone Pattern | Spiral

Straight Line — Halftone Pattern | Straight Line

Graymap:

Graymap will fill the image with gray at three level lighter, normal, darker. Consult the manual for a more in depth explanation of using Graymap

## Three formats for saving an image

PICT

which is the default mode for AppleScan. PICT format can be used with other applications which support PICT.

MacPaint

Saves the image as a MacPaint file. MacPaint will save a 8x10 inch portion of the scanned image. The image is measured from upper-left corner. The area in dots per inch is 576x720 dots. The user should be aware that the image may be clipped because of the amount MacPaint will save.

TIFF (Tagged Image File Format)

This file format is used by some applications which, do not support PICT format. Be sure that if a image is saved in this format that an application exist to support TIFF format.

Other methods

• There will be times when only a small part of the image is needed, to avoid having to save the entire image and consuming a large amount of memory, try boxing or lassoing the image. Copy or cut the image and save it in the scrapbook.

• If the image to be scanned is smaller than the overall area, the scan area can be reduced so that, the scanner will only scan the desired image. This can be done by using the resize □ tool from the tools menu and moving the bars on the scan control panel to only scan the image. This can only be done in AppleScan.

# Levels of HyperCard Stacks

Consult the HyperCard manual to show the levels for HyperCard stacks, which will provide a more indepth discussion about the levels. But an understanding of the difference between background and card levels will imporve planning.

To understand background and card levels think of a photography studio. The photographer has a series of backgrounds, but only one for a particular shot, for example, a white background for passport shots. The photographer will take many photographs with the different subjects and only use that one background. The photographer can add to the picture by using props but the background remains the same way it looked before.

The same can be said about the two levels in HyperCard stacks, the Background level and the Card level. The background will remain the same throughout the stack, the change will occur on the card level. Each time a new card is created the background is the identical throughout the stack unless another background is incorporated into the stack. If you decide to change the field by changing the font, the style, or the shape the field will change on each card.

Field in a stack can be on the background and/or the card level. If the field is created on the background level, it will be on every card in the stack. Remember, by altering any background (field, picture, button), the change will occur globally in the stack.

# Creating HyperCard Stacks

The information that has been collected will provide the foundation for creating successful stacks, the next step is to create the stack(s). Before beginning, refer to the layout plan of the stack. The question should be what do I want the stack to look like? This means the physical representation. You can either custom design a stack or copy one from an existing stack. Look through some of the stacks to see if any of the designs are appropriate for the project or draw a rough sketch of the card design. Creating a stack design from scratch will take some time, but, with a little planning and a working diagram the design phase should move quickly. Below are some suggestions for creating a custom stack, followed by how to use existing stacks.

## Creating a custom designed stack

- Open HyperCard
- Choose New Stack from the File menu (the Dialog box will have a blinking cursor to alert the user to name the stack.
- Name the stack.
- Directly beneath the box is a checkbox which says "Copy current background", click in the box so, that the X will disappear. If the X remains in the check box, the

new stack will have the background design of the card that is currently on the screen.

- Click on button marked **NEW**, a new stack is created, displaying a blank card.
- Begin designing the background on the card layer, use the tools and patterns available for the design. Use various images, to enhance the design of the card. Refer to the list of on page 6 under Graphic Art.

If the design is not the appearance it should be create a new card, each card can be edited individually. If the card design was done on the background each new card would look alike, therefore in the beginning phases work on the card layer not the background layer.

The cards being created are on the card level so that the background card will remain a blank template until the appropriate background design is chosen.

When the design has been completed, it has to be put on the background layer in order to be used throughout the stack.

- Select either the lasso or the box tool, pull down the menu **Paint** choose **Select All** this will capture the entire image on the card and place it into the clipboard. Cut the image and choose **Background** from the **Edit** menu and paste the image. Now you have created a background for the stack, each new card that is created will have the same background design.
- Delete the other designs from the stack by pulling down the menu and choosing Delete Card. This deletes the card from the stack. On the first card, you will be unable to delete the card, because it is the last card in the stack. Choose the lasso tool from the Tools menu, then choose Select All from the Paint menu, this will select the image on the card. Press the delete key, the image will disappear.

Using an existing background?

- Begin by locating a background suitable for the new stack
- Follow steps 2-3 for creating a stack
- Leave the X mark in the check box "Copy current background" this indicates to HyperCard to create a new stack using the existing background that is displayed on the screen.
- Click on the NEW button, a new stack should appear with the background that was chosen

Don't use any background?

- Follow step 1-4 for creating a new stack
- A new stack should appear with a blank screen.

Using more than one background for a stack?

It is possible to have several backgrounds within one stack. By using several backgrounds, the stack appears to be branching to other stacks. The user is able to follow the flow of a program by seeing changes on certain cards. The developer also has a chance to mix background and have one background remain independent of another. The procedure for this is similar to creating a new stack, in this case the developer copies a card and then pastes the card on the background layer of the stack. After doing the developer can check Stack info from the Objects menu, the dialog box should state the number of backgrounds in the stack. The dialog box will also state the size, name and number of cards within the stack. The new background will be independent of the other background and can have separate background buttons, pictures and fields. It is a good idea to create all the cads for a particular background at one time. Uniformity for the stack increases, each new background will be grouped together, allowing for easier accessibility for the developer.

### Background fields, buttons and pictures

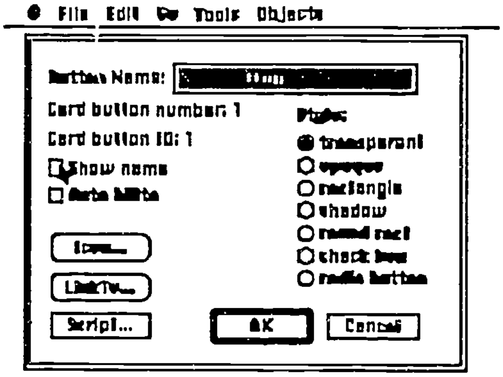
Previously there was a short discussion on the differences between card layers and background layers. While laying out the design for a stack, some cards will always have the same attributes, for example, the left and right pointing arrow. The right arrow will go to the next card and the left arrow to the previous card. For such functions it is easier to create the buttons on the background level. This enhances consistency throughout the stack, the buttons are always in the same position and performs the same function each time the button is clicked. An example, each card in the stack will have a button which will go the the menu card. The position of the button should be on the top right side of the card. The button should perform the visual Venetian blinds at all times.

This information provides a lot of guidelines

1. The button should be on every card.
2. The button should be in the same position at all times
3. The button performs a unique function (Venetian blinds).

This indicates to the developer that this button needs to be created on the background layer.

- To begin open an existing stack or create a new stack.
- Choose from the Edit menu, **Background**. This changes the appearance of the screen showing, two rows of slanted lines on the top of the menu bar, this will indicates that you are in the background layer.
- Choose New Button from the Objects menu. A rounded rectangle should appear called New Button. Double clicking on the button, opens a dialog box, which allows for changing the appearance of the button.



Button Name: [illegible]

Card button number: 1

Card button ID: 1

□ Show name

□ Auto hilite

Style:

● transparent
○ opaque
○ rectangle
○ shadow
○ round rect
○ check box
○ radio button

**Two choices:**

You can use an existing button, (one created by the developers of HyperCard) or use a picture and place a button on top of the picture.

### Choice One

To use an existing icon, click on the bı tton **Icon**, another dialog box opens with a listing of the available icons. Choose the icon and press the **OK** button. Notice that the button now has a icon.

### Choice Two

To have a blank button, click on the button **Transparent**, this creates a empty button. The checkbox called Show Name should be deselected, if left the name of the button will show. Click on the **OK**, this will return you to the background layer. The button should have moving dashes around it, move the button to the desired area on the card. Paste a picture on the background layer, positioning the picture over the button.

- Choose the button tool and double click on the button, choose Script from the dialog box; another screen will appear,

```
on mouseUp
|
end mouseUp
```

this is where HyperTalk commands are entered. At this point the commands should be keyed in

```
visual effect Venetian blinds to card
go to card id 8345
```

After completing the above steps return to the card layer, by pulling down the menu Edit and choosing Background, this deselects the background mode. The card level should have the background button showing, each time a new card is created a that button will appear in the same position and perform the same function.

The same rules will apply for creating fields and pictures on the background level. The choice is up to the developer, on deciding what will be on the background level ,ıd what will be on the card level.

## Sound and Animation

To use or not to use? The use of sound and/or animation depends heavily on the type of audience the stack is being designed to instruct. It has been my experience that the use of sound intensifies the computer-person relationship. Sound acts as an enhancement, to show praise, or to catch the attention of the user. Sound can also be used effectively for introducing a new project, such as opening a new stack, using music.

Good luck and happy stacking!