DESIGN STANDARDS AND CRITERIA FOR GRAPHICS USED BY PUBLISHERS AND MANUFACTURERS IN THE DEVELOPMENT OF EDUCATIONAL SOFTWARE

By

CARL DAVID PAYNE JR.

Bachelor of Science
Oklahoma State University
Stillwater, Oklahoma
1971

Master of Arts
Oklahoma State University
Stillwater, Oklahoma
1972

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
DOCTOR OF EDUCATION
May 1989

DESIGN STANDARDS AND CRITERIA FOR
GRAPHICS USED BY PUBLISHERS AND
MANUFACTURERS IN THE DEVEL-
OPMENT OF EDUCATIONAL
SOFTWARE

Thesis Approved:

____
Thesis Adviser

____

____

____

____
Dean of the Graduate College

FORWARD

"There is a real problem about the future, and we don't admit it. We assume we can see into the future better than we really can. Leonardo Da Vinci tried to make a helicopter five hundred years ago. From instances like that, we tend to believe that the future is predictable in a way that it really isn't. Because neither Leonardo or Jules Verne could have ever imagined, say, a computer. The very concept of a computer implies too much knowledge that was simply inconceivable at the time those men were alive. It was, if you will, information that came out of nowhere.

And, we are no wiser, sitting here now...we certainly can't guess what men might accomplish thousands of years in the future.

Michael Crichton

"Sphere"

ACKNOWLEDGEMENTS

I would like to thank the members of my doctoral committee: Dr. Kenneth King, Dr. Randall Koetting, Dr. William Rugg, and Dr. Bruce A Petty. I am especially grateful to Dr. Bruce A. Petty for his guidance, concern, and friendship during my graduate work at Oklahoma State University.

A special thank you is due to Maggie, whose courage endurance, patience, humor, and love sustained me when my confidence wavered, and whose support made the completion of this degree possible.

TABLE OF CONTENTS

Chapter | Page

CHAPTER I

INTRODUCTION

A scant ten years ago, says Electronic Learning (1987), the first microcomputer to be delivered to an American classroom was delivered to Arkansas High School in Texarkana, Arkansas. "That 8k, now long since traded in for a newer model, opened a dramatic era for education (p. 53)." Today, there is an increasing use of electronic technologies in teaching. Jerome Johnston (1987) noted several reasons for this trend: first, society is attaching a greater value to mastering new information and new skills; second, the urgency of meeting these needs, often by non-traditional means; and third, the evolution of information technologies themselves.

M.A. White (1988) has dubbed the emergence of the information technologies as the "third learning revolution." He contends this may very well bring about new and more radical changes in how we think than either the first revolution, the alphabet, or the second revolution, the printing press. Since the 1950's with the introduction of the technology of television, modern students have been introduced early to, and have mastered quickly, information delivering technologies. They develop a visual literacy that is a group of vision competencies fundamental to learning (King et al., 1984). Specifically, the impact of computers on society has parents demanding that schools prepare their children for the future technological market place (Tatenbaum and Mulken, 1986).

In order to effectively implement microcomputers in the classroom, educators must have a more complete understanding of the microcomputer's potential (Roth, 1987). To improve and enhance the potential learning experience, money is needed in three areas--teacher training, computer acquisition, and software development (Reed and Sauter, 1987).

In an era of increasing teacher accountability, it is important for teachers to have the same tools available to them as are available to professionals in other fields (Simen, 1988). These new tools affect learning and teaching which can change the very nature of how our minds work and, therefore, how we educate (White, 1988).

A survey of Electronic Learning Magazine (1987) reports that, while none of the fifty states requires all teachers in the state to take a computer course, thirteen states plus the District of Columbia do require all students (or others seeking teacher certification) to take a computer course. New York and West Virginia are strongly advising that their state teachers pursue computer courses. In two positive moves towards computer literacy in 1985, North Carolina required teachers seeking certification to be computer competent and, beginning in 1988, Wisconsin has required all teacher training institutions to make it possible for their students to become computer competent (Electronic Learning, 1987).

Since the introduction of the microcomputer a decade ago, its numbers and acceptance have continued to grow. Ingersol, Smith, and Elliot (1986) estimate that there were approximately one million microcomputers in the public schools in 1986. In fact, Bork (1986) estimates that the number of microcomputers in the public schools doubles every two years. Confirming Bork's estimate, The Sunday Oklahoman (1988) reported that there are two million computers in the nation's public schools. It is estimated that 95 to 96 percent of the nation's schools have at least one microcomputer (Ingersol, Smith, and Elliot, 1983).

What factors make possible this kind of growth in the number of microcomputers? R.B. Otte (1984) reports that one reason is the decreasing costs of the hardware with, in some cases, manufacturers offering systems for less that $1000 dollars. Electronic Learning (1987) reported some other reasons for the continued growth of computers in the classroom:

1) Integrating computers into the curriculum becoming a standard practice;
2) Better products being produced and growing confidence in those products;
3) Better prepared teachers, more knowledgeable

and more capable of implementing computer education programs;

4) Legislation making greater monies available;

5) Private and public sectors providing assistance.

Additionally, 80 percent of all states, plus the District of Columbia, recommended that schools provide students with exposure to computers (Electronic Learning, 1987). To this end, the U.S. State Department of Education reported estimated expenditures of $550 million for computer hardware and an estimated expenditure of $130 million on accompanying software in the 1985/1986 school year, and they expect increased expenditures for the 1987/1988 school year (Bozeman and House, 1988).

Anticipated software expenditures in twenty-eight states are expected to exceed expenditures for hardware (Electronic Learning, 1987). McGinty, Reed, and Sauter (1987) report that software buyers have some 600 software companies offering them products, an astounding 30,000 to 40,000 computer software programs for personal computers (Miller, S. K., 1987).

Twenty-eight of the fifty states responding to the Electronic Learning Survey (1987) report that their 1986/1987 software budgets will increase 5 to 25 percent. These expenditures and acquisitions figures for software are somewhat misleading. Research indicates that, while there is an overwhelming enthusiasm among teachers for computers in the public schools, these same teachers feel that a majority of these software products are of extremely poor quality (Hasset, 1984; Roth, 1987). Heckinger (1982) reported software programs were frequently boring and pedagogically flawed. Many educators feel the software is dull and unimaginative (Ingersol, Smith, and Elliot, 1983), while Ladd (1988) contends that one program in ten now on the market is capable of doing the job it states that it can do. Perhaps reflecting their dissatisfaction with total availability and quality, the Electronic Learning Survey (1987) reports that seventeen states currently have software development projects underway.

Educators who acquired computers with the expectations that they would be educational cure-alls have been largely disappointed (Tetenbaum and Mulkeen, 1986). Perhaps these educators did not know or forgot that computers are a demanding tool. Unlike most technology, the computer has no single purpose and, because it can be used in so many ways, it can be badly misused (McArthur, 1987). Holste (1988) contends that the microcomputer is both the problem and the solution. The success of computer instruction, contends Johnston (1987), rests more with the design of the instructional software program than it does with the computer itself.

One of the big problems in computer software design is that too much software is engineered by computer programmers or created by content experts. That not only creates problems, it creates bad software (Ingersol, Smith, and Elliot, 1983; Collis and Gore, 1987; Ladd, 1988). Many programmers, especially P.C. programmers, do not take software design seriously enough, believing that it requires little work or thought. The results are usually commercially acceptable, but contain inconsistencies and sloppy elements that should not be there (Reed and Sauter, 1987). Collis and Gore (1987) contend that the complexities of good software design are too great even for content experts using any one of the excellent authoring systems.

While many in education acknowledge that problems exist in current software design, they seldom concur on just what the problem(s) may be. Perhaps the first and greatest software design problem lies in the mind set of the designers, says Jerome Johnston (1987). He posits that electrical text and electronic graphics are the electronic version of printed text. What is needed are well researched production tools, or as Anderson (1983) might state it, design theories to be used to generate production practices.

E.F. Ferguson (1977) in his article, "The Mind's Eye: Non-Verbal Thoughts in Technology," points out that the educational system, including the designer, ignores the visual requirements of the new technologies. The language of technology is basically a non-verbal one, and people involved in this technology, whether it be film, video, T.V. or a monitor, need to think and design in terms of visual images (Ferguson, 1977; White, 1988). The evolution of software development has a history of less than 25 years (Johnston, 1987). It is now time to scrutinize computer software in terms of its substance, pedagogy, and visual imagery.

Significance of the Study

Teachers and administrators overwhelmingly encourage and support the use of microcomputers in the classroom; computers are proving to be one of the greatest technological breakthroughs in education to date (Grabawski and Aggen, 1984). Educators, however, continue to be disenchanted and dissatisfied with the software available for use with computer-aided delivery systems (Steffin, 1983; Tyler, 1983; Gold, 1984; Bork, 1987; Roth, 1987). Computer-aided educational delivery systems are those systems which exploit the use of the computer to assist educators improving productivity (Bunderson, 1982).

Educators need to look beyond the fancy packaging, the well-phrased usage directions, and the glitz of fast paced, high powered graphics...educators maybe seduced by the sirens of technology, warns Norman (1987). The problem, says Tatenbaum and Mulkeen (1986), is that software packagers rarely draw upon educational or psychological principles, such as advanced organizers, intermittent reinforcers, or feedback. All too often, the tenents of good instructional design, perception, and attention are also overlook or ignored. The deficiencies result in too much software that is inappropriate or technically unsound (Gold, 1984).

In 1984, Bork called for a great improvement in the materials available for use with microcomputers, and, by 1987, Bork was still contending that people had failed to produce much good computer software compared to educational needs. What is lacking in too many of the products from software designers is the understanding of the impact of all the parts of the software upon the user-learner. Allessi and Trollop (1985) call for authors to specify the purpose or goals for their software designs, while Ladd (1988) calls for software designers to meet the need and the existing skills of the audience they are designing for. Software developers and manufacturers need to be aware, specifically, of how software stimulates or reinforces learning (Williams and Williams, 1987).

It is important that people involved in software design adhere to a process which reflects both instructional design and software engineering principles (Williams and Williams (1987). These principles, or tools, as Norman (1987) would call them, are important for good software design.

In "The Cognitive Approach to Computer Courseware Design and Evaluation," Timothy Jay (1985) identifies five human information processing abilities which cognitive psychologists feel must be accounted for in good software design:

1) Cognitive characteristics of the user;
2) Feedback to the user;
3) Language and text characteristics of the user;
4) Memory and attention;
5) Graphic perception and visual processing.

Visual thinking pervades all human activity, from the abstract and theoretical to the down-to-earth and everyday (McKim, 1972). Fleming and McKim (1987) point out three reasons why designers should be aware of the perception principles:

1) It is important to avoid misconceptions. If a learner misunderstands the content of a paragraph or visual, they may learn something else.
2) When you must replace the real work with some graphic representation, it is important to know how to represent reality adequately.
3) The better an object, event, or relationship is perceived, the better it is remembered.

Reports abound in the literature regarding the lack of design criteria for good quality software. There are, however, few reports regarding the practices that software designers and developers use to improve the quality of their software designs. This study is intended, not only to make known the educational levels of the software designers, developers, and producers, but to discern their use of certain principles of attention and perception in their microcomputer software designs.

Purpose of the Study

The purpose of this study is to ascertain the education and experience level of those persons responsible for the design and development of microcomputer software as well as to investigate their use and manipulation of some important principles of attention and perception in their microcomputer software products. In this study, the important principles of attention and perception are those enumerated by Professor Emeritus Malcolm Fleming in his chapter entitled "Displays and Communication" in Robert Gagne's new book, Instructional Technology: Foundations. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher. These principles were derived from Fleming's latest research on attention and perception.

Limitations of the Study

The study has been limited to the software publishers identified in July of 1988 by the 1988 Educational Preview Guide published by the California Software Evaluation Consortium. The Consortium had identified these publishers, geographically located in the United States and Canada, as publishing at least one piece of high quality computer software worthy of consideration for preview by American educators.

Definition of Terms

Traditional Instruction: Methods and materials commonly used by most teachers in the public schools wherein textbooks are the primary source of information and the direction and delivery of that information is controlled and presented by the teacher to all students as a group (Roth, 1987).

Computer Assisted Instruction (CAI): Classroom instruction that is a combination of traditional instruction integrated with remedial, supplemental, or enrichment activities delivered to students through interaction with a preprogrammed computer (1987).

Computer Based Instructional (CBI): Classroom instruction in which the primary source of instruction is through interaction with a preprogrammed computer for individuals or small groups (Roth, 1987).

Computer Managed Instruction (CMI): A classroom activity that enables students and teachers to use the computer as a tool, i.e., word processing programming languages, and record-keeping, all to help guide students through their learning experiences (Allessi and Trollop, 1985).

Hardware: The physical computer equipment, including such items as the computer, the keyboard, the disk drive, the monitor, the telephone modem, the mouse, and the printer (Culp and Nichols, 1986, 1986).

Software: A learning package with self-contained programming or a code that drives or tells the computer what to do when loaded into the computer. It is the software that gives the computer its versatility (Allessi and Trollop, 1985).

Interface: A point or means of interaction between two systems, disciplines, or groups. In this instance, the term is synonymous with software, in that it is the mechanism with which learners interact when working with a computer (Faiola and DeBloois, 1988).

Frame: In programmed instruction, an individual picture or one picture in a series of pictures, whose vertical and horizontal limits are determined by the lines of resolution of the monitor; a block of verbal/visual information (Fleming, 1987).

Screen: An individual picture in a series of pictures, whose vertical and horizontal limits are determined by the lines of resolution and/or the physical limits of the computer monitor; a block of visually presented electronic text and/or graphics.

Instructional Development: The process of analyzing needs, determining what content must be mastered, establishing educational goals, designing materials to help reach the objectives, and trying out and revising the program in terms of learner achievement (Heinich, Molenda, and Russell, p. 439).

Instructional Technology: A complex, integrated process involving people, procedures, ideas, devices, and organization, for analyzing problems and devising, implementing, evaluating, and managing solutions to those problems in situations in which learning is purposive and controlled (Heinich, Molenda, and Russell, p. 439).

Technology of Instruction: A teaching/learning pattern designed to provide reliable, effective instruction to each learner through application of scientific principles of human learning (Heinich, Molenda, and Russell, p. 444).

Instructional Designer: A person responsible for carrying out and coordinating the planning work; competent in managing all aspects of the instructional design process (Allessi and Trollop, 1985).

Instructional Developer: One who manages the instructional subject area specialists, planning, development, and implementation procedures for instruction or training (Allessi and Trollop, 1985).

CHAPTER II

REVIEW OF RELATED LITERATURE

The purpose of this chapter is to present a review of literature to the following concerns:

1. impact of computers in education;
2. state of computer usage in today's public schools;
3. current state of software design, especially in regard to education/and training;
4. educational backgrounds of instructional software designers;
5. important considerations for instructional design;
6. important guidelines for effective software design;
7. designing effective software to sharpen and increase perception.

Computer technology is changing rapidly, both in the area of hardware and in the software that make computer use in education a reasonable pursuit. Effective design of such software and knowledge of effective learning strategies are crucial to effective use in educational settings.

Overview of Computers in Education

In a complex society dependent upon instantaneous communications and information, television and computers are rapidly becoming our dominant cultural tool for selecting, gathering, storing and conveying knowledge in representational forms (Adams and Hamm, 1988). Jerome Johnston (1987) contends the lure of electronic media lies in the promise of capturing the very best of instruction--the stimulating call to learn, the lucid explanation, the vivid portrayal of complicated phenomenon, the analysis and remediation of learning errors, and the repeated use. Quite simply, computers have permanently altered our environment, impacting it daily by information-providing technology (Faiola and De Bloois, 1988).

Computer history can be viewed in terms of progress toward making computers smarter, faster, smaller and cheaper (Miller, 1987). Modern computers come in three sizes: mainframes, mini and micro. In general, these sizes correspond to the amount of information and complexity of application that a computer can process (Miller, 1987).

Computer-assisted instruction was first developed in the 1960's on large mainframe computers (Johnston, 1987). A crude technology, by today's standard, has become a sophisticated tool of the artist, researcher and educator. It is no longer an inanimate tool, but an intellectual and active partner that, when fully exploited, can produce new intellectual forms and experiences (Patten, 1988). Modest costs and thousands of computer software programs are available to support education applications (Otte, 1984).

Appropriate implementation of microcomputer in instruction can provide a wide range of new experiences and thinking atypical of the convergent style of thinking in traditional education (Steffin, 1983). Effectively used, computers can not only transmit information but also help learners effectively process that information (Johnston, 1987). Additionally, while most systems in the real world are highly complex, the computer is the first medium able to transmit and model the complexity to its user (Pezdek, 1987). Because of these capabilities, Bill Honig, Superintendent of Public Instruction for The California State Department of Education, feels that computers are a vital component necessary to teach students to communicate, write and analyze (Reed and Sautter, 1987).

That computers can be an effective training medium for at least some situations is no longer in doubt (Holste, 1988). Also, no longer in doubt is education's commitment to increase computer resources in the schools. In a 1983 article, Becker (1983) reported 53 percent of the elementary and secondary schools had at least one computer. Williams and Williams (1985) reported that it was difficult to find a school without at least one computer. Electronic Learning's 1987 survey found that the number of microcomputers in elementary and secondary schools grew 18 percent between 1986 and 1987, and that total dollar expenditures for both hardware and software will remain at the 1986-1987 levels through 1987-1988 (Electronic Learning, 1987). To date, some two billion dollars has been spent on instructional computing this decade as a way to help students learn (The Daily Oklahoman, 1988). The increased spending has had effects on both teachers and students.

A 1987 Electronic Learning survey found that 80 percent of the states, and the District of Columbia, recommend that all students take a computer literacy course. More often, the recommendation from the remaining states is that computers be integrated into the traditional curriculum (Electronic Learning, 1987).

Jerome Johnston (1987) contends that, for the learner, the computer has great "intrinsic motivational qualities" that make it appealing in a way that text is not. This intrinsic quality can be seen in students' expression of more positive feelings about themselves and school. Consequently, these attitude changes are apparently transformed into measurable behavior changes at every grade level, in any educational area (Fisher, 1984).

Good teachers welcome anything that will help them do more for their students (Bunderson, 1982). Computer technology can provide teachers with a workable means for adapting the creative process to suit the needs and individual ability level of all students (Gallini, 1983). Additionally, Kulick et al. (1980) reported substantial and highly significant differences in the amount of time necessary to instruct by conventional means versus computer-based learning. In education there is an increased interest in using computers evidenced by:

-the number of new and task-specific computer labs that are being set up throughout the country;

-the number of teachers and educational personnel requesting computer training (<u>Electronic Learning</u>, 1987).

In a move to protect and to increase the potential of their dollar investments in educational computing, thirteen states, plus the District of Columbia, require that all students in teaching programs (or others seeking certification) take a course in computer topics (<u>Electronic Learning</u>, 1987).

The State of Computers in the Schools Today

Computers vary greatly in their capabilities (Kleiman, 1987), and any decision in developing a computer system involves trade-offs (Berger, Pezdek, and Banks, 1987). In the simplest sense, most common computer systems may be viewed as four units connected electronically:

1) An input unit (Keyboard), through which data is entered;

2) A processor unit, which stores data input and processes it electronically;

3) An output unit (computer terminal screen, or printer) which shows the results of processing the data input;

4) A data Storage/retrieval unit which stores data on and retrieves data from some magnetic medium (such as a floppy disk) (Culp and Nickles, 1986).

Surprisingly, all this fancy, expensive hardware has a half life of less than two years (Bunderson, 1982), after which it is no longer considered state of the art.

The capability of hardware has, in general, ceased to be a bottleneck in the development and use of CAE (computer assisted education) delivery systems (Bunderson, 1982). Yet, design problems for computer-based instruction are the same as print-based programmed instruction. So, why is computer-based instruction flourishing? One explanation is the appeal of the microcomputer as a tool. Another explanation is the intrinsic motivational characteristics that make it more appealing than text (Johnston, 1987). Finally, the computer-based assisted education is grounded in the studies of behavioral psychologists. Thus, people who are rewarded for a good response will continue that good behavior (Laird, 1985).

Computers are a natural for any subject that involves teaching about a complex system (Pezdek, 1987). Besides being fun, computers can build a bridge on which drudgery can become entertainment, and entertainment can become learning (Laird, 1985). This so-called fun, in general, has the great strength of allowing users to interact with complex systems having multiple interactive dynamic variables (Pezdek, 1987), CAI is significantly more effective in achieving higher performance among both low and high ability learners (Fisher, 1983).

Despite what would seem to be glowing reports on CAE, The Office of Technology Assessment, an arm of Congress, reports that the computer age is struggling to achieve little more than a passing grade in the nation's schools, which are plagued by a lack of funds, ineffective teacher training, and mediocre software (The Sunday Oklahoman, 1988). In a survey conducted by The Center for Social Organization of Schools, it was estimated that schools need one computer for every twelve students to provide thirty minutes of computer time a day for all students (The Sunday Oklahoman, 1988).

Two-thirds of the nation's fifty states, however, report that insufficient funding has hampered implementation of new technology, particularly computers. The medium ratio of students to computers ranges from a high of 42:1 (Brandon, 1988) to 30:1 nationally (The Sunday Oklahoman, 1988). Not all students in the schools use computers; those students who do use computers spend an average of only about four percent of their computer time on instructional use (The Sunday Oklahoman, 1988).

Brandon (1988) reports that the average time per week spent on computers by elementary kids is thirty-five minutes, by intermediate level students, sixty-five minutes, and by secondary level students, 105 minutes. In order to reduce the student-computer ratio to 3-to-1 it would cost a whopping four billion dollars, which would comprise an enormous chunk of the current instructional budget of the nation's public schools (The Sunday Oklahoman, 1988).

The computer is not going to replace the teacher in the classroom. It can, however, be an important addition to the curriculum by broadening it and by organizing and making available a tremendous number of materials for instruction (Reed and Sautter, 1987). Kansas Computer Education Specialist, Craig Haugsness, reports that many schools in his state now have three separate labs--one for business, one for instruction, and one for general purpose computing, mainly word processing (Electronic Learning Survey, 1987).

Expanding computer sections in Kansas may well be an exception. Nationwide, only one in three new teachers has had even ten hours of computer training (The Sunday Oklahoman, 1988). Jerome Johnston (1987) contends a teacher's role is to urge the learners to develop and utilize appropriate learning strategies. To enable teachers to do this, they should be able to pick the best software available to meet their needs (Simon, 1988).

Williams and Williams (1985), however, state that most teachers do not know how to find the software they need. Mostly, they locate software by word of mouth, workshops, inservices or demonstrations.

That computers can be an effective training medium is no longer in doubt. Open to discussion, however, are questions of who and when and what for and how much--and, increasingly, which of several variations of the technology is best to teach with (Holste, 1988). Bunderson and Inouye (1987) contend that the present delivery system will not solve the problems of work and knowledge in an economical way. The present delivery systems are near or at maturity level; thus, it will be difficult to increase academic productivity qualitatively, and the real deficiencies of the delivery system cannot be easily changed (Bunderson and Inouye, 1987).

On the other hand, there are those who feel that change can be effectively and intelligently managed. Jolicourt and Berger (1988) recommend the following steps to effectively and intelligently bring educational software into the classroom:

1) Specify overall goals of the implementation process;
2) Select the appropriate software;
3) Develop the necessary software support materials;
4) Assign students to comparable ability groups;
5) Schedule and implement computer time to students;

6) Test the students regularly;

7) Evaluate the success of your software and its implementation procedures.

Current State of the Computer Software

CBI (computer based instruction) refers to a curriculum program in which there is interaction between the student and the computer. Delivery of instruction is provided in the form of drill and practice, tutorial, simulations, games or problem-solving software disks (Bozeman and Housem, 1988). Information is transferred between a disk and a computer via a disk drive, which is a mechanism that holds the diskette and causes it to rotate at about 300 revolutions per minute. Some disk drives are built directly into the computer, while others are extended to the computer (Miller, 1987). Programs can be stored on magnetic tape, cartridges, hard disks, cassette tapes or floppy disks. A typical single-sided double density 5 1/4 inch disk can store the equivalent of fifty-five double-spaced typewritten pages (Miller, 1987).

Recently announced is the revolutionary new optical disk that can be erased or modified. To emphasize the enormous storage potential of the optical disk, the NeXT Computer System comes with an optical disk that includes all of the following: a full dictionary, a thesaurus, a book of quotations and the complete works of William Shakespeare (Magrid, 1988).

Many programs are nicely packaged, attractive and entertaining (Ignatz, 1985). They are meant to be used with high resolution screens, three dimensional mice, eye movement detectors, voice in, voice out, touch in and feelers out; superficial pleasure, but are not of any lasting value (Norman, 1987). The major problem, contends Bork (1987), is that technology is poorly used, the software much more so than the hardware.

Software design and development has long been a problem because of its poor design and inadequate educational values (Roblyer, 1983; Tyler, 1983; Hasset, 1984; Kamouri, 1984; Williams and Williams, 1985; Roth, 1987; Orlando, 1988). This on-going problem of software development has long been a barrier to CAE's widespread use (Bunderson, 1982). The problem is due mainly to programmers with little educational background or educators with little programming expertise (Gold, 1984). These instructional software designers, be they programmers or educators, are overly influenced by programming languages and the linear nature of the computer (Brandon, 1988), and they have a tendency to over-simplify the information to be conveyed (Ullner, 1988). The result has left many educator disappointed with the amount and quality of the software available (Steffin, 1983; Gold, 1984; Roth, 1987).

The development and production of software is slow and expensive (Brandon, 1988). To date, the most expensive software has been developed only when federal seed money was available (Naimen, 1988). Brandon reports that the development lag of business software which follows the introduction of new operations systems is often as much as eighteen months, and he suspects the lag for educational software is even greater. Development and production costs can run as much as $500,000 for simulations and $1,000,00 for some year-long courseware (Brandon, 1988). Publishers are not confident about the instructional software market, and, because of widespread piracy, hesitate to invest heavily in necessary development strategies or necessary personnel (Naimen, 1987).

Miller (1987) divides software into three distinct categories. The first category, tutorial, uses the computer to improve student performance in a particular skill or content area. The second category, simulations and games, uses the computer to model the environment, i.e., problem solving. The third category enables both teacher and students to use the computer as a tool, i.e., recordkeeping, word processing, and programming languages.

It is imperative that instructors consider the characteristics of the target population for whom instruction is being delivered (Dick and Carey, 1978). Major differences exist in the software needs of different subject-matter areas and grade levels. In early elementary grades, say Williams and Williams (1985), software is most commonly used to reinforce basic learning skills. In second through forth grades, software seems especially effective for introducing new concepts and creating new experiences. In the upper elementary grades, software can be used effectively as a tool for problem-solving and for programming. In high school, effective software includes the familiar productivity tools of word processing, spreadsheets, data base management, graphics and packages that introduce programming languages (Williams and Williams, 1985).

Ladd (1988) insists that "good software quiets restless audiences, engages their minds and stimulates their response (15)." Marge Kasel recommends that teachers should look for three main things when they consider whether or not to use a particular piece of hardware:

-Does the software use the computer well;

-A good piece of software should have an "easy in" and "easy out";

-Does the software have a lasting value (Shalvay, 1988).

Given the potential for software to manipulate text, graphics, color and sound, let alone to offer interactive qualities to the user, the question should be asked about the degree to which different educational software programs take advantage of these qualities (Williams and Williams, 1985). We are past the age of novelty, and we should be at the point of intelligent, credible design.

Instructional Design

Good courseware plays to its audience as surely as a good actor plays to his audience. Consequently, identifying the needs and characteristics of the people who will be using the program is the first step in courseware development (Ladd, 1988).

Good courseware, software, if you will, is no accident. It is the product of well conceived and well executed instructional development. What is instructional development? Bratton (1983) describes "instructional development" as the basic, systematic process of delineating instructional need, designing instructional solutions in light of state learner objectives, implementing the chosen solution and evaluating the results in terms of learning outcomes.

It is recommended that instructional developers have the following skills and competencies:

- -good interpersonal communication skills;
- -ability to conduct a needs analysis;
- -conduct task analysis;
- -formulate goals and objectives;
- -assimilate chunks of new information and work them into a logic framework;
- -use relevant principles of learning;
- -select and sequence content;
- -select appropriate instructional delivery systems;

-solve problems related to the instructional development;

-use multiple evaluation techniques;

-be able to work with diverse organizations, human beings and instructional constraints (Wallington, 1981; Phillips, 1981).

And finally, instructional designers are those individuals who can carry out the above processes in cooperation with a subject matter specialist (Bratten, 1983).

For computer based training and development, these additional competencies and skills are suggested:

-conduct feasibility assessment for computer based training;

-have a broad understanding of design principles for courseware development;

-possess ability to trouble-shoot courseware problems;

-prepare specifications for the courseware production;

-integrate any off-line materials if necessary;

-be a theoretician;

-be an instructional writer;

-be a media specialist;

-know authoring systems;

-know authoring languages;

-know programming languages;

-be able to analyze system configuration;

-be able to analyze software characteristics;

-be able to analyze systems for performance.

(Kearsly, 1983; Laird, 1985).

These are clearly complicated skills and involved tasks that would seem to move the development of well designed computer software beyond the capability of a teacher, content expert, and/or programmer. There are, however, many organizations in which one person is the entire operation (Laird, 1985). David Stone, premiere author/designer at Computer Teaching Corporation, says that producing good software takes a lot of time and money, and you need a good, skilled team of people to pull it off (Ladd, 1988).

Ladd contends that because designers and authors can take up to 280 hours to create each hour of instruction, the climb to proficiency for software designers is a long one and an expensive one, and, if Laird is correct, cost estimates range from "$2000 to $20,000 per instructional hour (p. 194)." This may well be, says Ullner (1985), why software designers have a tendency to define and design complex and difficult systems in simple terms. It may also explain why, even though software developers may have clear goals, they rely less on clear principles of software development and more on intuition to reach their goals (Karen, 1987).

The microcomputer in any educational setting is only a tool. What sets the computer apart from the pencil is its greater potential for expanded use. This potential is provided by the software, the better designed the software, the greater the microcomputer's potential. Good screen design can often represent successful interface between man and machine. Thoughtful utilization of text and graphics has proven to: (1) aid insight and understanding of relationships between concepts, and (2) be valuable as an illustrating process (Faiola and De Bloois, 1988). With good software, "microcomputers reduce learning time and improve the retention factor; they are not only 'wise', they do their work well (laird, 1988)."

Software Designers

A review of software packages reveals that the computer screen is frequently treated as an electronic page, with designers using the same rules and heuristics for both the design and the layout as that of printed information (Morrison et al., 1988). Branden (1988) finds that instructional software designers often continue to use outdated instructional methods in both printed and media formats. Compounding these mistakes, Grabiner and Albers (1988) report that the instructional design is also influenced by a publisher's decision as to the lay-out design.

Ladd (1988) posits that good software benefits from plain old common sense. People read from left to right and top to bottom. Don't try to be high-tech creative and interfere with people's natural tendencies. Quality software design requires quality in both the visual design and the educational design (Collis and Gore, 1987; Ladd, 1988). It is the task of the designer to establish a conceptual image of the system that is appropriate for the task and class of users (Norman, 1987). It is important that designers realize that users of software have their special needs (Norman, 1987). As Computer use increases, Kingdom (1988) sees human engineering factors becoming more important.

Morrison et al. (1988) reminds us that computer displays are dynamic in that they command attention. Because of the dynamic qualities of computer displays, there are no simple answers, only trade-offs; each application of a design principle has its strengths and its weaknesses (Norman, 1987).

Experienced and capable courseware authors are hard to find. The skills are complex and exacting. It takes at least two years to develop a competent author (Ladd, 1988). It is the designer's role, say Faiola and De Bloois (1988), to provide a friendly medium between the learners and the electronic learning environment. This means, among other things, that the program is easy to operate, facilitates learning of content without any wasted effort, and does not require any wasted effort on learning how to run the program (Hazen, 1985).

It is the designer who interprets, designs, and communicates information through the use of such visuals as graphics, videos, text, icons, and other visual clues (Faiola and De Bloois, 1988). Designers must sort through, understand, arrange and then apply information to be learned to an educational model (Ladd, 1988). It is the designer's responsibility to use learning and communication principles and visual research findings when they design their computer screen displays (Faiola and De Bloois, 1988). All of this is done, says Ladd (1988), with the underlying concern for the trainee's comfort; reading a computer screen is harder than reading material on paper.

Today's computer users are demanding better and more sophisticated software (Koenig, 1987). Roblyer (1983) says we need to make a considerable investment to facilitate change in the way courseware is currently being created. To not use all that we have learned to date about effective development, teaching, and learning methods would be missing an opportunity that may not come again (Roblyer, 1983). Kraft (1987) reminds us that the failure to employ proper strategies does not result so much in incoherent presentation, but often in so much less effective ones. Good design requires careful development and delivery of the text and graphics in a logical, coherent, and quality-consistent relationship of components in the visual field in order to improve the user's comprehension (Orihuela, 1986). It is, after all, say Williams and Williams (1985), the computer program [software] that supplies the learning experiences.

Considerations for Software Designers

Good software design has an easy in, easy use, and easy out. The software designer needs to avoid making excessive demands on the working memory of the user. The learner will be unable to work in the "short term memory mode" with more than four or five items, steps or operations (Gagne and Glasser, 1987). Since this memory has only a limited capacity, it cannot be loaded beyond that limit. If that limit is approached, the following is likely to occur:

- a slowing down of the processing; i.e., it takes more time to understand the meaning of the information in the exercises;
- a deterioration in the visual interpretation of the representation; e.g., the ignoring of parts or all of the presented information.

(Kline, 1985).

As long ago as 1972, Dwyer (1972) was reporting that the brain is capable of utilizing only minute portions of the information perceived. If this is true, the task then, for software designers, is to find an efficient and effective visual schema for the presentation of both the text and graphics within the microcomputer's frame.

In whatever sense or way that learning happens, education from a computer requires that the learner be placed in front of a pre-programmed learning sequence (Holste, 1988). Johnston (1987) would remind the software designer that computer-based instruction first requires a sophisticated task analysis, not to mention that the designer needs to remember that the microcomputer's language is basically a non-verbal one. Thus, he needs to think in visual terms.

The trick to good visual design is to keep it simple. Too much detail often means students spend too much time reading (or processing) the visual. If a concept is that complex, the visual design may be better seen and studied on paper (Ladd, 1988).

The secret of good design, according to Nelson (1981), is the dedication of the software designer in adhering to the principles of visual design. A well designed grid (screen) improves comprehension and shortens readers' search time (Faiola and De Bloois, 1988). Software designers should think of the software layout, both text and graphics, as a dynamic spatial and temporal instructional design (Faiola and De Bloois, 1988).

Good software needs to be consistent. It needs to be easy enough to use by the beginner yet not burdensome or repetitive for the expert to use (Kingdom, 1988). The software designer needs to organize the elements in the design into coherent units which, in turn, is organized into a unified pattern (Nelson, 1981). At all times, maintains Kingdom (1988), the user needs to have a visual indication of where he is in the program. For Ladd (1988), if the design of the text and the graphics "doesn't enhance the student's education, it's no good no matter how clever it is" (p. 21).

Designing Visual Perception

In general, instructional technology contributes knowledge that aims to improve an individual's learning, mastery and competence (Kaufman and Thegarajan, 1988). Instructional systems, maintain Tennyson and Park (1986), represent attempts to make and direct predictable connections between instructional variables and learning outcomes. In order to meet these goals, says Kemp (1985), an orderly, logical method of identifying, developing, and evaluating learning strategies is needed.

A critique of contemporary instructional design theory, among other things, reports Numan (1984), posits that there are two basic types of instructional design: (1) those produced by teachers, and (2) those produced by professional instructional designers. Johnston (1987) says that, regardless of who develops the instructional design, it is essential to remember, "learners differ in their intellectual curiosity--their innate desire to engage new material--and their skills at decoding messages from various media" (p. 20).

The visual sense is highly discriminating, constantly analyzing visual elements to accept or reject new information based upon what it already knows (Anderson, 1983). There is general agreement, say Adams and Hamm (1988), that, to be fully literate today, children must understand print and electronic images. Adams and Hamm go on to say that, because these electronic symbol systems are central to visual communication and to thinking, their interaction and interdependence cannot be ignored. Recent research, says Pezdek (1987), suggests that all sorts of computer skills are quite dependent on visual abilities. Attention is highly selective; we can give attention to only a small part of the environment at one time, and of that, we see most sharply only the tiny central portion of the visual field (Treisman, 1974; Fleming, 1988). Even then, seeing an image does not automatically guarantee learning from it (Adams and Hamm, 1988).

Software and Perception

De Graff (1985) states that there is no purposive behavior without perception. He goes on to say that behaviorist instructional design is grounded in the assumption that people learn from what they perceive. A learner's perception of a particular concept may vary depending on the method used to define the information; i.e., words or visuals (Fleming, 1988). Therefore, De Graff (1985) says, it is important to remember, "perceptual stimuli can be modified to influence behavior such as learning in a positive way."

Learning is an active process within an individual, requiring that the learner willingly engage the material (Johnston, 1987). Learning is facilitated where the learner reacts to or interacts with the crucial information, and the more activity, the more learning, within limits (Fleming, 1988). Kaufman and Theagarjan (1988) contend that instruction is a means for closing gaps in performance.

Good design comes from a combination of intelligence and creativity (Nelson, 1981). The success, therefore, of training depends on how wisely and how well we design and deliver learning opportunities (Kaufman and Theagarajan, 1988). It is an essential aspect of good instruction to ensure that the student attends to the important information (Allessi and Trollop, 1985). A designer's whole function, contends Nelson (1981), is to create an arrangement of visual elements in any creation or design which will satisfy the human need for both order and variety.

The manipulation of a visual presentation, says Klein (1985), is built upon the assumption that the perceptual system responds in almost stereotypical ways to characteristics of the environment. Fleming and Levie (1978) posit that perception is not registered in absolute values, but functions by comparisons, and is based in part on the learner's expectations. The visual system seeks out simple, geometric regularities, including parallelism, and uses them to resolve ambiguities in perception (Butler and King, 1987). Fleming (1988) says that probably the most important perceptions by learners are the various kinds of symbols used in instruction, for instance, words, numbers, pictures, which refer to important phenomena from their environments. Thus, successful design strategies are those that reinforce and facilitate the reading perceptual cycles (Grabing and Alberns, 1988).

Aust (1988) reminds us that central theories of perception contend that the perceptual aspects most crucial to learning are the self-directed processes which enhance the interpretation of stimuli. Displays that are organized reduce the possibility that the learner will organize the material differently or perhaps erroneously (Fleming, 1988 and Grabinger and Albe, 1988). Learners try to construct meaningful wholes from their environments, say Eysneck (1984), and unorganized stimulation is difficult to understand and remember. Orderly displays should invite systematic perceptual processing (Fleming, 1988) and do it in an easy to follow manner (Nelson, 1973).

Klein (1985) posits that there are visual patterns which have special informational value. Human being are sensitive and attentive to the spatial relationships among parts (Klein, 1985). Thus, it is important, says Fleming (1985), to separate the visual field into figure and horizon and use a recognizable standard to help determine size or quantity of an object.

A directional pattern should be evident (Nelson, 1981). There is an evolutionary sensitivity to motion and changes within the environment (Klein, 1985). Fleming (1988) recommends using a reference point, such as the frame or background, to thus be able to judge the rate or direction of any motion. Finally, Metallones and Chartrand (1987) would remind software designers to be aware of and avoid the use of rapid inward/outward movement of visual elements as they decrease the viewer's ability to receive, process, and recall detailed information.

The perception of color is a complex interaction of physiological and psychological factors (Gordon, 1988). Color can be used to add interest, create spatial dimension, create moods, separate or emphasize elements, or identify objects by their associative color, i.e., apples are mostly red (King et al., 1981). Fleming (1988) states that vision is most sensitive to colors in the middle of the color spectrum, yellow and yellow-green, and least sensitive to colors at either end of the spectrum, violet-blue and red.

Another aid to software users' perception is sound. The female voice commands more attention from the human mind than does the male voice (WGBH, 1988). Hearing, says Fleming (1988), is more attuned to mid-level pitches than to pitches at either end of the sound spectrum. Pezdek

(1987) posits "that the presence of the audio track actually improves comprehension of visually presented information" (p. 11).

Software and Attention

The relative value and educational worth of microcomputer software is heavily dependent upon the visual design of both text and graphics within the picture plane and/or the picture frame. King et al. (1981) define picture plane as the actual surface on which the designer creates the visual and picture frame as the boundary of the picture plane. This design of displays that communicate is not a stand-alone process; rather, it grows out of prior analysis of learner characteristics, tasks, and learning situations. Quite simply, states Fleming (1988), without attention, there can be no learning. It is the designer's job to keep attention focused on the important information (Allessi and Trollop, 1985 and Brody, 1982).

Learner expectations can strongly influence attention (Eysneck, 1984). Expectations, says Salomon (1984), can also influence the amount of mental effort that learners are willing to invest in attention to a visual display. The objective, contends Orihuela (1986), is to make the most important parts of instructional visual the center of the learner's interest.

Hartly (1982) says,

> Given a particular text and a particular page, one can manipulate the spatial arrangement of the text on the page so as to enhance clarity, retrieval, and comprehension (p. 196).

This flow or 'spatial movement' facilitates a clear organization and helps the reader quickly read the page (Orihuela, 1986). This manipulation of the visual design of the software is crucial. Fleming (1988), maintains that perception, in particular, is most influenced by the informative areas of the display; informative areas are those most effective in inducing and providing a match between display structure and learner schema. Sautterthwaite (1984) also recommends using the principle of stimulus differential to direct the attention of the software user to 'devices' that are sufficiently different from other components of the design to ensure that those devices will be noticed.

The design of the page or screen, contend Grabinger and Alben (1988), suggests something about the content on the page or screen. Ladd (1988) notes that the idea is to keep attention focused to the content. He recommends a low density level of information, placing only a few sentences of text or visuals on each screen display. Fleming (1988) suggests software designers attempt to design visual displays that are moderate in their nature. Allessi and Trollop (1985) caution against visual designs that are too easy or obvious because they will fail to hold attention. The density level, the richness or detail presented in each visual screen display, should not overwhelm the viewers

(Morrison, Rose, et al., 1988). Graeser, Lang and Eafson (1987) recommend that the screen density level of a microcomputer display should not exceed 25 percent. By using this leaner design strategy, Fleming (1988) maintains that the software designer can focus attention on the relevant information.

Ladd (1988) contends that software designers should be consistent and put the crucial information in the same quadrant of the screen so users do not have to hunt around for commands and information. The more pertinent information, recommends Fleming (1988), should be placed in the central portion of the viewing field. Ladd (1988) warns that if every screen is a surprise, learners become quickly frustrated; to avoid frustrating the learner, he recommends that software designers avoid the following:

- -Long text blocks
- -Surprises
- -Clutter

It is important to consider not only the pictorial content of a display but the pictorial form as well (Kraft, 1987). The better organized or patterned a message is perceived, the more information is available for learners to receive and process at one time (Fleming and Levie, 1978). Attention is drawn to what is novel or different; by manipulating the displays, the software designer can readily introduce novelty (Berlyne, 1970).

Fleming and Levie (1978) maintain that novelty can be used as a stimulus to influence learner attention. This need not be entirely new to the learner, it only needs to be different from what the learner has just recently experienced (Berlyne, 1970). Fleming (1988) maintains that novelty can be further understood as the introduction of change.

When attention is gained, change in the ongoing stream of information can help maintain it. Vision, says Fleming (1988) is especially sensitive to change across space. For example, Metallonis and Chartrand (1987) state that a shot or visual representation approaching a viewer arouses greater interest than does a shot or visual representation moving away from the viewer. Francis and Evans (1987) report that blonde models are more favorably perceived than brunettes, especially if they are in the same visual. And, changing the brightness perceptably of one screen to the next also attracts and maintains attention (Fleming, 1988).

In those systems that are capable of sound, there are several important considerations. Fleming (1988) recommends a change in the volume and/or the inflection of speech. People respond more favorably to information presented by a female voice (WGBH, 1988), and they are more attentive to sound that is one or two octaves above or below middle C (Fleming, 1988).

Software designers should change the pace of the action or instruction within the frame and change the rate of the action or instruction among succeeding frames, recommends Fleming (1988). The depths of field within screens are used as reference points to help viewers establish psychological reference points and to help then perceive and to comprehend information (Metellonis and Chartrand, 1987). These judgements on distance or depth are relative and inversely related. The smaller the size of an object, says Fleming (1988), the further away it is perceived. Software designers need to also keep in mind that the ability to perceive, process, and make judgements on the visual presentation within a frame is analogous to the speed at which the images move in/out of the visual space/frame/screen (Metallonis and Chartrand, 1987). This is why attribute isolation, or including only the most relevant information, becomes of extreme importance (Bovy, 1987). Pezdek (1987) recommends increasing the time the visual or information is on the screen or in frame from the usual average of eight seconds to as much as sixteen seconds for the benefit of both younger and older learners when you are not dealing with a fixed screen/frame.

Color is another device that can attack attention. Dwyer and Berry (1982) suggest that color aids viewers in gaining a more complete awareness of the content presented. Reed and Sautter (1987) recommend using color only as an accent to or with the visuals present. With color, says Koening (1987), the software user can recognize the information more readily. Color may also increase the information capacity of a visual display (Allessi and Trollop, 1985).

Color, maintains Allessi and Trollop (1985), can be easily misused so as to be ineffective or even detrimental to learning. Dwyer and Berry (1982) warn that stylized or unrealistic coloring of instructional visuals are usually effective only with students of higher intelligence. Software designers should limit the use of 'hot' colors and never use too many colors of any kind (Kingdom, 1988; Hazen 1985; and Faiola and De Bloois, 1988). If a software designer uses too many colors, warns Kingdom (1988), it becomes unclear what the colors are trying to indicate. Color has the ability to change the entire condition and atmosphere of a visual design, not to mention its ability to motivate and evoke a large range of psychological responses (Faiola and De Bloois, 1988).

Words and text can attract attention. Instructional text, maintains Garbinger and Albers, (1988), encourages an interaction between the learner and the text. Readers, warn Nelson (1981), do not have time to learn a new alphabet or reading scan pattern. Kemp (1980) recommends the use of simple, easy-to-comprehend lettering styles. Use large letters, printed in upper case (all capitals) rather than a mix of upper and lower case letters (Nelson, 1981; Grasser, Lang, and Eafseon, 1987). Kemp (1980) also recommends using a minimum of lettering styles, to maintain simplicity for visual reading.

Morrison et al., (1988) suggest to software designers that they minimize the amount of text to be presented. Text is easier to comprehend when it is broken into meaningful units as opposed to arbitrary windows (Yeaman, 1987). Anderson (1983) also reminds software designers that, if words are visually displayed they need to be clear, and viewers need enough time to read them.

For the most part, states Hartley (1978), there are four kinds of sentences:

-Simple

-Compound

-Complex

-Compound-complex.

Oriheula (1986) maintains that because simple sentences contain less information, they are easier to understand; therefore, paragraphs should be constructed with an easier sentence form and kept as short as possible, especially when directed to younger learners.

Orna (1983) reports that readers have an easier time with concrete rather than abstract words, and ordinary, everyday words rather than special words. Readers also prefer a variety of sentence types between the personal and the impersonal; readers will prefer active sentences to passive sentences if the subject is the receiver of the action. Otherwise, passive sentences are generally easier to understand (Orna, 1983; Orihuela, 1986). Finally, Fleming (1988) reminds software designers that beyond using concrete words, concrete nouns more readily elicit mental images than do abstract nouns.

Nesbit (1987) posits that individual visuals themselves have the greatest influence on looking behaviors. The ability to focus attention is improved with cues (Brannon and Williams, 1987; Fleming, 1988). Garner (1987) states there is evidence that location cueing improves the learner's performance on task. Examples of effective cues are: arrows, underlines, circles, rectangles around items, inverse lettering, captions for visuals, and verbal directions (Bovy, 1987; Fleming, 1988). Kirk and Gustafsen (1986) feel color also may be used constructively to cue attention, while Levie and Lentz (1982) suggest that pictures without some attention-directing cues may be scanned superficially and processed on a very shallow level.

Ladd (1988) maintains that whether a software user will learn or not depends on how well the visual elements work on the screen. Attention, perception and learning, suggests Fleming (1988), are not discrete processes; they are richly intertwined. Words and images reinforce each other, say Sutherland and Winn (1987), appealing to both verbal and visual memories. Every choice, reminds Kraft (1987), may affect the viewer's understanding and subsequent memory of the visual.

Contrary to common assumptions, a medium is not a neutral transmitter of information, suggests Greenfield

(1987); through its formal and technical characteristics, a medium transforms information while communicating it. For the instructional designers, the goal is, says Orihuela (1986),

> To create a very simple, unified piece of instructional material, in which the components are arranged in the best way possible to convey the required information (p. 5).

Williams (1987) reports that there is no formula for making software, but there are critical elements in making a successful piece of software. Each piece of software requires new content, confronts new learners, and has new needs to fulfill. Visual information design should be consistent with and integrated into the rest of the instructional message (Allessi and Trollop, 1985). Thus maintains Ladd (1988), good courseware design is clear, telling students where they are, how far they've come, and where they are going; it helps the student avoid feeling like they are in a maze.

To help instructional software designers, Ladd (1988) recommends using the following principles of visual presentation:

-Good visual presentations are legible

-Good visual presentations are accurate

-Good visual presentations are colorful

-Good visual presentations are simple

-Good visual presentations are unified.

When the software designer communicates better, the software user processes better (Allessi and Trollop, 1985).

Summary

Computer history can be summarized as an attempt to make computers smarter, faster, cheaper, and smaller, with size corresponding to complexity of application and total amounts of information processable.

From a crude technology in the early 1960's, Computer Assisted Instruction has become a sophisticated tool. It has the potential to become an intellectual and active partner with a learner, and is capable of providing new experiences and thinking atypical of the convergent style of thinking prevalent in education today...when designed properly.

To date, some $2 billion has been spent on educational computing. Students, for the most part, are motivated to use computers. Unfortunately, most teachers are currently ill-prepared to effectively use the computer in their classrooms, and most lack the knowledge base to intelligently find and appraise the software packages available for their curricular needs.

Software design has long had a problem with inadequate educational values and poor instructional design. Too many software packages are overly influenced by programming languages, the linear nature of the computer, and an over simplification of the information presented. Much of the software available today fails to use the computer well and allow for an easy in and out to the program, and has not represented a lasting educational value for the money invested.

Software that is the product of a well conceived and well executed instructional design identifies and fulfills the needs and characteristics of its users. It should represent a successful interface between machine and man, reducing learning time and increasing retention.

A great many software packages are being designed with the same rules and heuristics as have been used in the layout and design of printed matter. Software designers must sort through, arrange, and then construct the thematic content in accordance with current learning theory and visual communication rules, always mindful to avoid making excessive demands on the learner's working memory.

Since the computer's basic presentation form is essentially a non-verbal format, designers should think of the layout of both text and graphics as a dynamic, spatial and temporal design. The visual sense is highly discriminating, and all sorts of computer skills are dependent upon acceptance or rejection of the visual elements.

Behaviorist instructional design assumes that people learn from what they see. Perception of educational information is dependent upon how willingly learners do or do not engage the material presented. Thus, the software designer's task is to create designs that willingly engage the learner's perception in a way that satisfies the learner's need for order, variety, and a meaningful whole from the information.

Relationships, movement, color and sound all have the ability to positively or negatively impact a learner's willingness to engage and learn from the software; this is also true for movement, cueing devices, novelty elements, and spatial arrangement when limited to the informative areas of the picture frame.

A medium is not a neutral transmitter of information; it transforms information while communicating it. Thus, how well a software user learns is due to how well the software design performs on the screen. Words and visuals reinforce each other, appealing to and stimulating both verbal and visual memories, and they most often affect the user's understanding and subsequent memory of the software's information.

There are no 'easy' formulas for software design. The array of talent, education, and skills necessary for good software design are often beyond the capacity of a single individual. What is needed is a developmental mind-set that embraces, as a matter of curricular philosophy, an approach to software design that takes advantage of a software development team. Such a team should consist of a content expert, an instructional designer, a graphic designer, a computer programmer, and a classroom teacher grounded in educational computing.

CHAPTER III

DESIGN OF THE STUDY

This study was meant to assess the education and experience level of those persons with the responsibility for the design and development of a manufacturer's computer software.

Additionally, this study investigated those designers' use and manipulation of some important principles of attention and perception in the development of computer software.

In this study, the design criteria for the software evaluation were selected from the chapter "Displays That Communicate" by Indiana Professor Emeritus Malcolm Fleming, written for Robert Gagne's book, Instructional Technology: Foundations, Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher. Fleming's principles were derived from numerous reviews of current research on attention and perception (Appendix A).

Subjects

The subjects of this study were microcomputer software publishers and their instructional designers and developers. These microcomputer software publishers were identified by The 1988 Educational Software Preview Guide. The subjects were located in the United States and Canada. There were a total of 105 subjects identified by the editorial staff of The 1988 Educational Software Preview Guide at the time of the survey (Appendix B). By the time The 1988 Educational Software Preview Guide was published in the fall of 1988, twenty-seven additional software producers were identified, but they were not included in the survey. The subjects had been identified by The 1988 Educational Software Preview Guide reviews board as producers of high quality software.

Procedures

These microcomputer software publishers were contacted by letter (see Appendix C) asking that they fill out a survey requesting information about the education and/or expertise levels of the instructional designers or developers who created microcomputer software for them. The survey instrument also asked for information concerning the use and manipulation of certain principles of attention and perception by the instructional designers or developers who created microcomputer software for them (Appendix D).

If no response was received in a period of four weeks, a follow-up letter was sent (Appendix E) requesting that they return the survey.

If no response was received on the follow-up request within ten days, an attempt was made to telephone the subjects to request that they return the survey. Some of the survey participants answered the survey over the telephone.

Treatment of the Data

The data from the survey participants were categorized and percentages were calculated to provide the percentages for educational levels, areas of study, experience, and the use of the principles of attention and perception. The following are the criteria listed on the survey:

1) For software that your company has determined to publish, who has the creative responsibility for the development of the visual design and/or layout?

-freelance writers

-company designers

-the publisher

2) What is the educational level of the person with the creative responsibility for the development of the visual design and/or layout of your software?

-High School

-B.A./B.S.

-M.A./M.S.

-EdD./PhD.

3) Number of courses or hours of training in:

-Computer Programming (study of languages, programming retrieval and storage systems)

-Educational Psychology (study of the nature and operation of human learning.

-Instructional Design (study of systematic designs for instruction, educational materials, and technology)

-Educational Computing (study of computing in the design, analysis and presentation of instruction)

-Graphic Design (study of the art/science of presenting the printed word or illustrations)

-Visual Design (study and design of materials that appeal directly to sight, i.e., TV, Film, advertising)

-Other

-Unknown

Attention

1) Do your software designers create software that places the most pertinent information:

-evenly over the entire visual field?

-over the central portion of the visual field?

-over the upper one-third of the visual field?

-over the bottom one-third of the visual field

2) Do your software designers attempt to design visual displays that are:

-of simple complexity?

-of moderate complexity?

-of extreme complexity?

3) Do your software designers attempt to focus the user's attention through visual displays that are:

-lean - includes only the most relevant information?

-moderate - includes relevant information and a degree of realism?

-realistic - attempts to approximate reality?

4) Do your software designers manipulate instructional displays by: (yes, no, occasionally, comments)

a. changing the volume of speech?

b. changing the color or type face of your print?

c. changing the pace of the action or instruction?

d. changing the rate of the motion of the visual field?

e. changing the brightness of the visual field?

f. inducing some change in the ongoing stream of instruction once attention has been gained?

5) Do your software designers attempt to guide attention by using the following cues: (yes, no, occasionally)

-arrows

-underlinings

-circles

-rectangle about items

-inverse lettering

-verbal directions

-captions for visuals

-other

6) Software designers often attempt to induce careful attention by manipulating the degree of uncertainty in the visual display. Do your designers:

A. when working with concept questions attempt to achieve:

-no uncertainty in the program?

-moderate uncertainty in the program?

-a lot of uncertainty in the program?

B. when working with problem-solving questions attempt to achieve:

-no uncertainty in the program?

-moderate uncertainty in the program?

-a lot of uncertainty in the program?

Perception

7) Perception, as defined here, means to grasp mentally, or to become aware of thorough sight, hearing, or touch. With this in mind, please respond to the following. Do your software designers: (yes, no occasionally, comments)

a. Separate the visual field into figure and horizon?

b. Give more attention to key figures, objects or persons than the background scene?

c. Use a recognizable standard to help determine size or quantity of an unfamiliar object?

d. Use a reference point, such as the frame or background, to judge the rate and or direction of motion?

e. Use colors that are primarily in the middle of the color spectrum throughout the program?

f. Use sound pitches that are two or three octaves above or below middle C (normal)?

8) Who, in your company, is responsible for the final visual design and/or layout of the final published version of the software that your company decides to market (choose one):

-the software publisher?

-the software designers?

-the software programmer?

9) Education, major, and years of experience of the person responsible for the final creative design/layout of the software programs that you publish?

CHAPTER IV

ANALYSIS OF DATA

Participants in the Study

Of the 105 publishers of educational software identified by The 1988 Educational Software Preview Guide, fifty-one (or 49%) of the participants responded. Fifty-four (or 51%) either did not respond, refused to respond because of company policy or went out of business between the compilation of The 1988 Educational Software Preview Guide and the execution of the study. Thirty-four of the fifty-one participants responded by mail while seventeen of the participants responded to telephone contact.

The range of responses differed greatly. Of the respondents, thirty (or 59%) added additional responses to either qualify or clarify their answers on the survey. Two participants sent separate letters, one declining to answer the survey because they were only a dealer/distributor and the other a very detailed answer to each question.

The responses of the publishers were categorized by the researcher by question. Responses do not equal 100% because of multiple responses. The responses were computed into percentages answering each option and their comments were listed as follows:

Creative responsibility for the development of the visual design and/or layout?

	N	%
- free-lance writers	06	12
- company designers	24	47
- the publisher	09	18
- free-lance writers and company designers	07	14
- free-lance writer/company designer/the publisher	03	06
- failed to mark the survey	02	04

-Additional comments from the respondents:

-one person with most of the responsibility with occasional additional staff help

-team effort

-all people involved in the design are current or former teachers

-we have our own designers

-depends if the software is developed in house or submitted by a free-lance designer

-can be all of the above

-two people share the responsibility

-free-lance graphic artists

-company designers supervised by the publisher

-company designers who supervise free-lance designers

The educational level of the person with the creative responsibility for the development of the visual design and/or layout of your software? (check degree and list major subject area)

	N	%
-High School	00	00
-B.A./B.S.	10	20
-M.A./M.S.	17	33
-EdD./PhD.	12	24
-did not respond	02	04
-Other	10	20

-Major subject areas were listed as follows:

-17.6% Education

- 7.8% Curriculum and Instruction

- 5.8% Physics

- 5.8% Math

- 3.9% Engineering

- 1.9% Math Education

- 1.9% Math depending on product

- 1.9% Education Psychology

- 1.9% Applied Physics

- 1.9% Electrical Engineering

- 1.9% Reading Education

- 1.9% Music/Math Education

- 1.9% Instructional Technology

- 1.9% Art & Graphic Design

- 1.9% English/Communications

- 1.9% Science Education
- 1.9% Foreign Language
- 1.9% Analytic Philosophy
- 1.9% Computer Science
- 1.9% Psychology
- 1.9% Teacher Education
- 52% No response

Number of courses or hours of training in: (respond to areas that apply to you)

	N	%
-Computer Programming from 10 to 300 hours and/ or 1 to 20 courses.	24	47
-Educational Psychology from 9 to 100 hours and/or 2 to 60 courses.	27	52
-Instructional Design from 3 to 100 hours training and/or 1 to 12 courses	15	29
-Educational Computing from 10 to 300 hours training and/or 2 to 12 courses.	13	24
-Graphic Design from 6 to 50 hours training and/or 1 to 20 courses.	15	29
-Visual Design 3 to 100 hours training and/or 1 to 15 courses.	12	24
-Unknown	10	20
-Other	04	08
-Did not mark anything	03	06
-None of these courses	01	02

-Hand written comments:

-many hours of educations and years of training

-lots of science/physics courses

-courses in logic, communications and writing

-Engineering

-Art

-Linguistics

Number of years teaching experience of the person with creative responsibility at:

	High	Low	Average
-Elementary	22	1	7.5
-Secondary	18	3	8.6
-City/County Vo-Tech	01	0	1.0
-Community/Junior College	22	1	8.9
-University/College	52	1	9.5
-Unknown			13.7
-Other			1.9
-Did not mark this question			5.8

Experience level of persons with creative responsibility for the design of the software:

	%
-Elementary	35.0
-Secondary	43.0
-City/County Vo-Tech	01.9
-Community/Junior College	01.9
-University/College	45.0

Software designers place the most pertinent information in the software design:

	%
-evenly over the entire visual field	50.9
-over the central portion of the visual field	15.6
-over the upper one-third of the visual field	13.7
-over the bottom one-third of the visual field	09.8
-unmarked	09.8

-Hand written comments:

-depends on the program

-whatever the designer chooses to use continually through the program

-all apply at different times

-combinations of all

-varies with different types of software

Degrees of complexity incorporated into your software visual displays:

	%
-of simple complexity?	58.8
-of moderate complexity?	31.3
-of extreme complexity?	01.9
-unmarked	07.7

-Hand written comments:

-try to be consistent and not go from one to the other in the same piece of software

-simple complexity is contradictory - poor question

-varies with the type of software

-depends on the age level

Software designers attempt to focus the user's attention through visual displays that are:

	%
-lean - includes only the most relevant information?	33.3
-moderate - includes relevant information and a degree of realism?	49.0
-realistic - attempts to approximate reality.	07.8
- Unmarked	09.7

Hand written comments:

-be consistent with choice

-varies with the software

-varies with the age group

Software designers manipulate institutional displays be:
(one check for each item please)

	Respondents %	Yes	No	Occasion-ally
a. changing the volume or inflection of speech?	92.0	09.8	64.7	17.6
b. changing the color or type face of your print?	100.0	54.9	25.4	19.6
c. changing the pace of the action or instruction?	94.0	39.2	43.1	11.7
d. changing the rate of the motion of the visual field?	94.0	33.3	49.0	21.5
e. changing the brightness of the visual field?	96.0	19.6	54.9	21.5
f. inducing some change in the ongoing stream of instruction once attention has been gained?	94.0	45.0	39.0	09.8

-Hand written comments:

-unknown

-no sound being used

-depends on age level if we manipulate a change in type face or color

-this does not apply to 'tool' software

-pace under the user's control

-not sure what the question has asked

-Apple II computers don't have enough flexibility to manipulate most of these

-our programs are accounting software.

Software designers attempt to guide attention by using the following cues:

	Respondents %	Yes	No	Occasion-ally
-Arrows	92.0	54.9	19.6	17.6
-Underlining	96.0	39.2	39.2	17.6
-Circles	90.0	15.6	54.9	19.6
-Rectangles About Items	94.0	47.0	27.4	19.6
-Inverse Lettering	94.0	41.0	47.0	05.8
-Verbal Directions	90.0	31.3	52.9	05.8
-Captions for Visuals	90.0	37.2	43.1	07.8
-Totally unmarked question/option(s)	07.8	00.0	00.0	00.0
-Other (written comments)	31.1	27.4	03.9	00.0

Hand Written Comments:

-animation

-different color lettering

-apple display has limited display capability

-use all occasionally except inverse lettering

-flashing words or cursors

-glossary and help screens

-highlighting

-italics

-inverse lettering too distracting

-icons

-asterick.

Software designers attempt to induce attention by manipulating the degree of uncertainty in the visual or display.

	%
A. When working with concept questions attempt to achieve:	
-no uncertainty in the program?	47.0
-moderate uncertainty in the program?	35.2
-a lot of certainty in the program?	05.9
-Unmarked question.	09.8

Hand written comments:

-a lot of reality

-question is vague

-don't understand the question.

	%
B. When working with problem-solving questions attempt to achieve:	
-no uncertainty in the program?	47.0
-moderate uncertainty in the program?	43.0
-a lot of uncertainty in the program?	03.9
-marked two options.	05.8
-Unmarked question	09.8

Hand written comments:

-don't understand the question.

-use all possibilities at various time.

To help the software user grasp mentally, or become aware of through sight, hearing, or touch, software designers attempt to increase perception when they:

		Respondents %	Yes	No	Occasion-ally
a.	Separate the visual field into figure and ground.	96.2	35.2	31.3	19.6
b.	Give more attention to key figures, objects or persons than the background scene?	94.0	47.0	37.2	09.8
c.	Use a recognizable standard to help determine unfamiliar object?	82.3	43.0	29.4	09.8
d.	Use a reference point, such as the frame or back-ground, to judge the rate and direction of motion.	84.3	45.0	37.2	01.9
e.	Use colors that are primarily in the middle of the color spectrum throughout the program?	82.3	37.2	35.2	09.8
f.	Use sound pitches that are two or three octaves above or below middle C?	78.0	25.4	35.2	07.8
g.	Unmarked question	15.6	00.0	00.0	00.0

Hand written comments:

-use what's available.

-don't give any special attention in any of our software.

-use all, depending on target age group and type of software.

-rarely use sound.

-limited by colors available on computers.

-don't use sound.

Those responsible for the final design and/or layout of the final published version of the software that your company decides to market are:

	%
-the software publisher	58.8
-the software designer	21.5
-the software programmer	07.8
-the software publisher/the software designer	03.9
-the software publisher/the software designer/ the publisher	05.9
-Unmarked	01.9

Hand written comments:

-publisher: basically I make few changes

-publisher/designer/programmer are all one person

-team effort with publisher, designer, programmer

Education, major and years of experience of the person responsible for the final creative design/layout of the software programs that you publish.

Education reported as follows:

-High School	01.9
-B.A.	15.6%
-B.S.	15.6%
-M.S.	07.8%
-M.A.	15.6%
-EdD.	01.9%
-PhD.	13.7%
-no reported level	15.6%

Years of experience reported as follows:

-2 years	01.9%	15 years	03.9%
-4 years	01.9%	18 years	01.9%
-5 years	05.8%	19 years	01.9%
-6 years	09.8%	20 years	05.8%
-7 years	03.9%	22 years	10.9%
-8 years	03.9%	30 years	01.9%
-10 years	01.9%	45 years	01.9%
-12 years	07.8%	52 years	01.9%
-14 years	01.9%		

-high, 52 years; low 2 years; average, 14.6 years.

-Major areas of study were listed as follows:

-Education	19.6%	Music	1.9%
-Math	13.6%	Foreign Language	1.9%
-English	7.8%	Spanish Lit	1.9%
-Computer Sc	7.8%	Ed. Technology	1.9%
-Graphic Design	5.8%	Administration	1.9%
-Engineering Sc.	5.8%	Social Science	1.9%
-Physics	3.9%	Curriculum	1.9%
-Art	3.9%	Adult Ed	1.9%
-Math Ed	1.9%	Reading	1.9%
-Philosophy	1.9%	Bio. Chemistry	1.9%
-Science	1.9%	Electric Eng	1.9%
-Psychology	1.9%	Chemistry	1.9%

CHAPTER V

SUMMARY, CONCLUSION, AND RECOMMENDATIONS

Summary

The use by educators of microcomputers for instruction in the classroom continues to grow. The literature in the field is full of reports about microcomputer technology's ability to patiently drill students, improve hand-eye-brain coordination through games, expand or provide specialized information with tutorials, and to provide or simulate experiences not otherwise available with simulations. Yet, there are still grave concerns by educators and administrators over both the quantity and the quality of educational software.

Many educators who regularly use microcomputers in the classroom feel that the software has often failed to hold student interest, shown little imagination, and was of questionable pedagogical soundness (Ingersol, Smith and Elliot, 1983). Alfred Bork contends that the major problem with the technology (both the hardware and the software) is that it is poorly used. "We have inadequate hardware and software, the software much worse than the hardware by large (Reed and Sautter, 1987, p. 20).

Delaware State Specialist in Educational Computing, Collen P. Wozniak, bemoans, not the need for more software but for software in different curriculum areas (Electronic Learning, 1987). Alfred Bork, however, states that "people haven't produced that much educational software compared to educational needs (Reed and Sautter, 1987, p. 20)." A 1987 study by Susan Roth found that the great majority of software publishers in her study had no formal or set criteria for their own software design and development or for selection of software that they would publish which had been submitted to them from outside sources. In fact,

> Overall, the great majority of educational software publishers do not have a formal or standard set of criteria to guide in software development or employ in the selection of educational software for publication (Roth, 1987, p. 57).

This study assessed the education and experience level of those persons with the responsibility for the development, design, manufacture, and publication of educational microcomputer software. The study used criteria from "Displays That Communicate' by Indiana Professor Emeritus Malcolm Fleming for a chapter he contributed to Instructional Technology: Foundations, edited by Robert Gayne.

The 105 subjects were software designers, developers, and publishers identified by The 1988 Educational Software Preview Guide. These individuals were initially contacted by letter requesting that they return the enclosed survey covering their education and/or experience level and their use of certain tenets of attention and perception. The data from the surveys were classified and percentages were calculated in accordance with the replies.

Conclusions

The 1988 Educational Software Preview Guide identified 105 publishers whom they felt published quality software. Of those 105 publishers, fifty-one, or 49%, of the publishers responded. Thirty-four of the fifty-one participants responded by mail while the remainder were contacted via telephone.

The participants' data was analyzed and percentages calculated for their responses to each question. The percentages are as follows:

Forty-seven percent of the respondents reported that the creative design responsibility for the software published was from in-house designers, 12 percent of their software publications were submitted by free-lance designers (for which no educational data is available), 18 percent reported (publisher qualifications will be examined later), with the remaining percentages split between combinations of these three groups. Among the in-house designers with design responsibility for the software, 33 percent reported they had Master's degrees. Unfortunately, only 17.6% of the respondents reported, as a major field of study, Education, followed by Curriculum and Instruction (7.8%), Math (5.8%), and Physics (5.8%). For those who design software, what appears to be lacking in their academic backgrounds is a firm grounding in the tenants of Learning Theory, Instructional Design/Development, Graphic Design, and Educational Computing.

While 53% of the participants reported having course work in Educational Psychology, none reported having a degree in the subject area. This is also largely true for the areas of Computer Programming (47% report some course work, but 1.9% report a degree), Graphic Design (29% report some courses; only 1.9% report a degree) and Educational Computing (25.4% report some courses; only 1.9% report a degree).

It may well be that both the publishers and the designers feel some course work combined with teaching experience at the University/College level (45%), and/or secondary teaching experience (43%), and/or elementary teaching experience (35%) substitute as suitable prerequisites for software design. Their responses on the survey do not support this notion.

Clearly, the lack of a strong knowledge base in Instructional Design, Learning Theory, Graphic Design and Educational Computing has had repercussions in the design and development of efficient and effective educational software. Nearly 59% of the designers favored placing pertinent information evenly over the entire visual field instead of the recommended central position of the visual field. Despite research that 'simple' levels of information complexity turn off or bore students, 58.8% of the designers favor a simplistic information level to a more stimulating and attention retaining moderate complexity. Instead of using 'lean' screen displays to focus and retain attention, the survey participants (49%) favor using a moderate density level screen design.

Good software design should attempt to manipulate the screen's visuals to attract attention and increase perception. With the exception of changing the color of the type face (54.9% of the designers use this method) the participants overwhelmingly have failed to take advantages of other manipulative devices (where appropriate) such as increasing/decreasing the volume or inflection of sound or speech, or increasing decreasing the rate of motion in the visual field. Once again, where software designers have had an array of cueing devices available to them to gain attention and increase perception, only 54.9% had used Arrows as cues, while other devices, such as Inverse Lettering (41%), Rectangles About Items (47%), Underlining (39.2%), and Circle About Items (15.6%), go largely ignored. Other perception enhancing devices, such as separating the visual field in figure and ground (35%), providing a known standard or reference for comparison (45%), giving more attention to key figures/objects (47%), using colors found primarily in the middle of the color spectrum (37.2%), or using sound that falls within two or three octaves above/below middle C (25.4%), are just too seldom used.

The participants reported that the responsibility for the final design of the published software rested with the publisher (58.8%). While 23.4% of the publishers reported their highest level of education as having either a Master of science or Arts degree, only 19.6% reported their major area of study as Education, followed by Math (13.6), English (7.8%), Computer Science (7.8%), Graphic Design (5.8%), and Engineering Science (5.8%). Sadly, only 1.9% of the publishers report degrees in such areas of study as Curriculum, Educational Technology, and Psychology. Unfortunately, crucial areas of knowledge such as Learning Theory, Instructional Development/Design, and Educational Computing are not perceived to be part of a necessary knowledge base for editors, the very people who have the final say on the content design and quality of the software that gets published.

Recommendations

The recommendations that follow should not in any way be viewed as an exhaustive list of recommendations to be followed by the software designers. They do, however, delineate a group of considerations made evident by a thorough review of the literature and the responses to the survey used in this investigation, that should merit immediate consideration and addition into the work of microcomputer software designers and/or the publishing houses themselves.

1). Software publishers should give serious consideration to a team approach to software design that would include, (1) a content expert, (2) an instructional designer, (3) a visual designer, (4) a graphic designer, (5) a programming expert, and (6) a guest classroom teacher or school curriculum expert to test pilot the software.

2). Design criteria and procedures should be based upon the latest learning research and regularly updated.

3). Based upon that learning research, publishers should develop and distribute sound design procedures to their design teams for each of the types of educational softwares, i.e., tutorials, simulations, games, and drill and practice.

4). The design criteria and design procedures should be applied to free-lance software submissions.

5). Publishers should make more of an effort to guide attention and perception to help students focus on critical information.

6). Publishers should insist that the individual(s) who makes the final decisions concerning software design has a strong educational base in the discipline area for which the software is being developed, as well as in curriculum development, and <u>educational</u> computing.

Suggestions for Future Research

The possibilities for research concerning education, microcomputers, software, attention, perception, vision, and instructional design are unlimited. The research attempted was an overview of critical design criteria and procedures that should be used in the development of all types of microcomputer software. Before there can be any improvement on the design of the various types of computer software, the following research projects need to be completed:

1). Survey software publishers about their design procedures as concerns the use of attention in the following software types:

Game software
Drill and practice software
Simulation software
Tutorial software.

2). Survey software publishers about their design procedures as concerns the use of perception in the following software types:

Game software
Drill and practice software
Tutorial software
Simulation software.

3). Survey software publishers about their design criteria and procedures as concerns the use of the principles of learning, i.e., sequential information presentation, repetition or frequency of information patterns as used in the following software:

Games software

Drill and practice software

Simulations software

Tutorial software.

4). Survey software publishers about their design criteria and procedures as concerns the use of the principles of concept formation (the ability to generalize across varied sets of examples) in the following software:

Games software

Drill and practice software

Simulations software

Tutorial software.

BIBLIOGRAPHY

Adams, D, and Hamm, M. (1988, January) Changing Gateways to Knowledge: New media Symbol Systems. TECH Trends, 33 (1), pp. 21-23.

Alessi, S.M. and Trollip, S.R. (1985) Computer Based Instruction. Englewood Cliffs, New Jersey: Prentice Hall, Inc.

Anderson, R.H. (1983) Selecting and Developing Media for Instruction. New York: Van Nostrand and Reinhold Company.

Anglin, G. (1986, Fall) Prose-Relevant Pictures and Older Learners Recall of Writer Prose. ECJT, 34 (3), pp. 131-136.

Anglin, G. (1987, Spring) Effect of Pictures on Recall of Written Prose. ECJT, 35 (10), pp. 25-30.

Association for Educational Communications and Technology (1980) Competencies for Instructional Development Specialists. Instructional Innovator, 25 (9), pp. 27-31.

Aust, R. (1988) The Development of Intelligent Perception. Proceedings of Selected Research Paper Presentation at The 1988 Annual Convention of the Association for Educational Communications and Technology, New Orleans, LA.

Becker, S. (1983, September) How Schools Use Computers. Classroom Computer Learning, 3, pp. 41-44.

Becker, A.V. (1987) Proceedings of Selected Research Paper Presentations at the 1987 Annual Convention of the Association for Educational Communications and Technology, Atlanta, GA.

Berg, P. and Bramble, W.J. (1983, Fall/Winter) Computers and the Future of Education. AEDS Journal, 17 (1), pp. 101-108.

Berger, D., Pezdek, K. and Banks, W.P. (Eds.) (1987) Applications of Cognitive Psychology: Problem Solving Education and Computing. Hillsdale, NJ: Lawrence Earlbaum Associates, Publishers.

Berlyne, P.E. (1970) Attention as a Problem Solving Device. In D.I. Mostifsky (Ed.), Attention: Contemporary Theory and Analysis. New York: Appleton-Century-Cofts.

Bertleson, C.L. and Klenrin, V.W. (1988, February) Teaching Graphics for Communication. T.H.E. Journal, 17 (2), pp. 71-73.

Bialo, E.R. and Erickson, L.B. (1985, Summer) Microcomputer Courseware: Characteristics and Design Trends. AEDS Journal, 18, pp. 227-236.

Bonner, P. (1985) Does Computer Assisted Instruction Work? Computer and Electronics, 11, pp.44-47 and 92-93.

Bork, A. (1977) Learning with Computers. Bedford, MA.: Digital Equipment Corporation.

Bork, A. (1984, October) Education and Computers: The Situation Today and Some Possible Futures. T.H.E. Journal, 12, (10), pp. 82-97.

Bork, A. (1986, February) Education and Computers, The Situation Today and Some Possible Futures. T.H.E. Journal, 15, (2), pp. 92-97.

Bork, A. (1987, May/June) Visions of the 1990's. What Experts Predict for Educational Technology in the Next Decade. Electronic Learning, 7 (6), pp. 20.

Bozeman W. and House, J.E. (1988, February) Microcomputers in Education: The Second Decade. T.H.E. Journal, 17 (2), pp. 82-86.

Bovy, R.C. (1981) Successful Instructional Methods: A Cognitive Information Processing Approach. Educational Communications Technology Journal, 29, pp. 203-217.

Brandon, P.R. (1988, October) Resent Developments in Instructional Hardware and Software. Educational Technology, 23 (10), pp. 7-12.

Brannan, J.R. and Williams, M.C. (1987, June) Allocation of Visual Attention in Good and Poor Readers. Perception and Psychophysics, 41 (1), pp. 23-28.

Bratton, B. (1983) The Instructional Design Specialist-Subject Matter Expert. Educational Technology, 23 (6), pp. 13-16.

Brody, P.J. (1982) Affecting Instructional Textbooks Through Pictures. In D.H. Jonassen (Ed.), The Technology of Text: Principles for Structuring, Designing, and Displaying Text. New Jersey: Englewood Cliffs.

Brody, P.J. (1984) A Research-Based Guide to Using Pictures Effectively. Instructional Innovator, 29 (2), pp. 21-22.

Bunderson, C.V. (1982, February) Will Technology Improve Instructional Productivity. Instructional Innovator, 27 (2), p. 29.

Bunderson, C.V. and Inouye, D.K. (1987) The Evolution of Computer-Aided Educational Delivery Systems. In R. Gagne (Ed.), Instructional Technology: Foundations. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher.

Butler, D.L. and King, A.M. (1987) Integration of Features in Depictions as a Functional Size. Perception and Psychophysics, 41 (2), pp. 28-42.

Chartman, S. (1980) Story and Discourse: Narrative Structure in Fiction and Film. Ithaca: Cornell University Press.

Cohn, V.B. (1988, January) Computer Courseware: Development and Evaluation. Educational Technology, 23 (1), pp. 9-14.

Collis, B. and Gore, M. (1987, February) The Collaborative Model for Instructional Software Development. Educational Technology, 22 (1), p. 40-44.

Computer Teaching Lagging in Schools, (1988, September) The Sunday Oklahoman, p. 20.

Crichton, M. (1987) Sphere. New York: Ballantine Books.

Culp, G.H. and Nickles, H.L. (1986) An Apple for the Teacher: Fundamentals of Instructional Computing (2nd. ed.). Monterey, CA: Brooks/Cole Publishing Company.

Degraff, J.T. (1985) Towards an Analysis of Instructional Television Presentation Forms: A Case Study. Ann Arbor, MI: U-M-I Dissertation Information Service.

Dick, W. and Carey, L. (1978) The Systematic Design of Instruction. Texas: Scott.

Durret, J. and Tezena, J. (1982) How to Use Color Effectively: The Elements of Color Vision and Their Implications for Programmers. Pipeline, 7 (2), pp. 13-16.

Dwyer, F.H. (1972) A Guide for Improving Visual Instruction. State College, PN: Learning Services.

Dwyer, F.H. and Berry, L.H. (1982) Interactive Effects of Color Realism and Learners' IQ on Effectiveness of Visual Instruction. Perceptual and Motor Skills, 54, pp. 1087-1091.

Electronic Learning (1987, October) Educational Technology 1987: A Report on EL's Seventh Annual Survey of the States. Electronic Learning, 7 (10), pp. 39-47, 83.

Eysenck, M.W. (1984) A Handbook of Cognitive Psychology. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher.

Faiola, T. and De Bloois, M.L. (1988, August) Designing Visual Factors Based Screen Display Interface: The New World of The Graphic Technologist. Education Technology, 23 (8), pp. 12-18.

Fergeson, E.S. (1977) The Minds Eye: Nonverbal Thought in Technology. Science, 197, pp. 827-836.

Fisher, G. (1983) Where CAI is Effective: A Summary of Research. Electronic Learning, 3 (3), pp. 82-84.

Fisher, G. (1984) The Social Effects of Computers in Education. Electronic Learning, 2 (6), pp. 26-28.

Flake, J., McClintok, C.E. and Turner, S.V. (1985) Fundamentals of Computer Education. Belmont, CA: Wadsworth Publishing Company.

Fleming, M.L. (1980) From Seeing and Hearing to Remembering: A Conception of the Instructional Process. Instructional Science, 9, pp. 311-326.

Fleming, M.L. (1988) Displays and Communication. In Robert Gagne (Ed.), Instructional Technology: Foundations. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher.

Fleming, M.L., and Levie, W.H. (1978) Instructional Message Design from the Behavioral Sciences. New Jersey: Educational Technology Publications.

Forgus, R.H. (1966) Perception: The Basic Process in Cognitive Development. New York: Mcgraw-Hill.

Francis, S.K. and Evans, P.K. (1978) Effects of Hue, Value, and Style of Garment and Personnel Coloring of Model on Perception. Perceptual and Motor Skills, 64 (2), pp. 383-390.

Gagne, R. and Briggs, L. (1979) Principals of Instructional Design (2nd Ed.). New York: Holt, Rinehart and Winston.

Gagne, R. and Glasser, R. (1987) Foundations in Learning Research. In R. Gagne (Ed.), Instructional Technology: Foundations. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.

Gallini, S. (1983, April) What Computer Assisted Instruction Can Offer Toward Creative Thinking. Educational Technology, 3, pp. 7-11.

Garner, W.R. (1987) Location and Color as Cuing Dimensions in Contingent Classifications. Perception and Psychophysics, 41 (3), pp. 202-210.

Gold, P.C. (1984, Fall) Educational Software - New Guidelines for Development. AEDS Journal, 18, pp. 41-50.

Gordon, C. (1988) Color in Computer Graphics. A V Video, 10 (3), pp. 26-30.

Grabawski, B. and Aggen, W. (1984, February) Computers for Interactive Learning. Instructional Innovator, 29, (2), pp. 27-30.

Grabinger, R.S. and Albers, S. (1988) The Effect of CRT Screen Design on Learning. Proceedings of Selected Research Paper Presentation at the 1988 Annual Convention of the Association for Educational Communications and Technology, New Orleans, LA.

Graesser, A.C., Land, K.L. and Eafseon, C.S. (1987) Some Tools for Redesigning System-Operator Interfaces. In D. Berger, K. Pezdek an W.P. Banks (Eds.), Applications of Cognitive Psychology: Problem Solving, Education and Computing. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher.

Greenfield, P. (1987) Electronic Technologies, Education and Cognitive Development. In D. Berger, K. Pezdek and W.P. Banks (Eds.), Applications of Cognitive Psychology: Problem Solving, Education and Computing. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher.

Gueulette, D.G. (1988, January) A Better Way to Use Television in Our Classrooms. TECH Trends, 33 (1), pp. 27-30.

Halpern, D. (1987) Analogies as a Critical Thinking Skill. In D. Berger, K. Pezdek and W.P. Banks (Eds.), Applications of Cognitive Psychology: Problem Solving, Education and Computers. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher.

Hand, J.D. (1982) Brain Functions During Learning: Implications for Structuring, Designing and Displaying Text. In D.H. Jonassen (Ed.), The Technology of Text: Principles for Structuring, Designing and Displaying Text. NJ: Englewood Cliffs.

Hartley, J. (1978) Designing Instructional Text. London: Kogan Page.

Hartley, J. (1982) Designing Instructional Text. In D.H. Jonassen (Ed.), The Technology of Text: Principles for Structuring, Designing and Displaying Text. NJ: Englewood Cliffs.

Hassett, S. (1984) Computers in the Classroom. Psychology Today, 14, pp. 22-28.

Hays, D.S. and Birnbaum, D.W. (1980) Preschoolers' Retention of Televised Events. Developmental Psychology, 16, pp. 410-416.

Hazen, M. (1985) Instructional Software Design Principles. Educational Technology, 25 (11), pp. 18-23.

Heckenger, F. (1982, April) About Education: Computer Software Found Weak. The New York Times, p.4.

Heinich, R., Molenda, M. and Russell, J.D. (1985) Instructional Media, (2nd ed.). New York: Macmillian Publishing Company.

Helson, H. (1974) Current Trends and Issues in Adaptation-Level Theory. In P.A. Fried (Ed.), Readings in Perception: Principle and Practice. Lexington, MA: Health.

Hofstetter, F.T. (1981) Computer-Based Instruction: Roots, Origins and Application, Benefits, Features, Systems, Trends and Issues. Delaware: University of Delaware. (ERIC Document Reproduction Service (No Ed.), pp. 231-343).

Holste, P. (1988, September) High Tech - Hot Talk Computer-Based Training. A Special Report from Training Magazine. 25, pp. 5-8.

Horton, D.L. and Mills, C.B. (1984) Human Learning and Memory. Annual Review of Psychology, 35, pp. 361-394.

Huber, L.N. (1985, October) Computer Learning Through Piagets' Eyes. Classroom Computer Learning, 6 (2), pp. 29-43.

Hurt, J. (1987, June) Assessing Functional Effectiveness of Pictorial Representations Used in Text. ECTJ, 35 (2), p. 85-94.

Ignatz, M.E. (1985, Fall) Suggestions for Selecting Science Education Software. Journal of Computers in Mathematics and Science Teaching, 5, (10, pp. 27-29.

Ingersol, G. Smith, C. and Elliot, P. (1983, October) Microcomputers in American Public Schools: A National Survey. Educational Computer, 2, p. 28.

Jay, T.B. (1985, Fall) The Cognitive Approach to Computer Courseware Design and Evolution. Educational Technology, 23 (1), pp. 23-26.

Johnston, J. (1987) Electronic Learning: From Audio Tape to Videodisc. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher.

Jolicour, K. and Berger, D.E. (1988, September) Implementing Educational Software and Their Evaluating Academic Effectiveness, Part I. Educational Technology, 23 (9), pp. 7-13.

Kamouri, A.L. (1984, April) Computer-Based Training: A Cognitive Framework for Evaluating Systems Design. Journal of Technology Systems, 12, pp. 287-309.

Karen, P. (1987) Why Does Software Design Lag Behind Hardware Development. P C World, 12 (4), p. 35.

Kaufman, R. (1972) Educational Systems Planning. New York: Prentice-Hall.

Kaufman, R. and Thiagarajan, S. (1987) Identifying and Specifying Requirements for Instruction. In R. Gagne (Ed.), Instructional Technology: Foundations. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher.

Kearsley, G.P. (1983) Computer-Based Training: A Guide to Selection and Implementation. Reading, MA: Addison-Wesley.

Kemp, J. (1980) Planning and Producing Audiovisual Materials, (4th ed.). New York: Harper and Row.

Kemp, J. (1985) The Instructional Design Precess. New York: Harper and Row.

King, K.L., Ludrick, J.A., Post, G.L., Strickland, J.R. and Petty, B.A. (1981) A Systematic Approach to Instructional Media Competency, (4th ed.). Dubuque, Iowa: Kendal/Hunt Publishing.

Kingdom, K. (1988) Expert Views on the Human Interface Traits of Successful Commercial Software. Solution Systems: S. Weymouth, MA

Kirk, F. and Gustafsen, K.L. (1986) Instructional Technology: A Systematic Approach to Education. New York: Holt, Reinhart and Winston.

Kleinman, G.M. (1984) Brave New Schools. Resten, VA: Reston Publishing Company, Inc.

Kline, D. (1985) Conditions Influencing the Effectiveness of Animated and Non-Animated Displays in Computer Assisted Instruction. Ann Arbor, MI: U-M-I Dissertation Information Service.

Keonig, K. (1987, December) Keyboarding: Prelude to Composing with a Computer. English Education, 19 (4), pp. 244-279.

Kraft, R, (1987) Rules and Strategies of Visual Narratives. Perceptual and Motor Skills, 64 (1), pp. 3-14.

Kulik, J.A. and Kulik, C. (1987) Computer-Based Instruction: What 200 Experts Say. Precedings of Selected Research Paper Presentations at the 1987 Annual Convention of the Association of Educational Communications and Technology, Atlanta, GA.

Ladd, B. (1988) Four-Star CBI: It Takes People to Make the Machines Perform. Computer Based Training, 12 (5), pp. 14-18.

Laird, D. (1985) Approaches to Training and Development, 2nd Reading, MA: Addison-Welsey.

Levie, W.H. and Lentz, R. (1982) Effects of Text Illustrations: A Review of Research. ECTJ, 30 (4), pp. 195-232.

Lesser, G. (1972) Assumptions in the Production and Writing Methods in Sesame Street. In W. Schram (Ed.), Quality in Instructional Television. Honolulu: University Press of Hawaii.

Levin, J.R. (1981) On Functions of Pictures in Prose. In F.J. Pirozzola and M.C. Wittrock (Eds.), Neuropsychological and Cognitive Process in Reading. New York: Academic Press.

Mager, R. (1975) Preparing Instructional Objectives. California: Pitman Learning, Inc.

Markle, S. (1978) Designs for Instructional Designers. Illinois: Stipes Publishing Co.

Magid, L.J. (1988, October) Can NeXT Live Up To Media Hype? The Tulsa Sunday World, G-3, p. 9.

Martin, D.S. and Ditcham L. (1987, January) Information-Processing Analysis of Television Advertisement Recall. The Journal of General Psychology, 114, (1), pp. 9-12.

McArthur, D. (1987) Developing Computer Tools to Support Performing and Learning Complex Cognitive Skills. In D. Berger, K. Pexdek and W.P. Banks (Eds.), Applications of Cognitive Psychology: Problem Solving, Education and Computing.

McCarthy, R. (1988) Stop the Presses: An Update on Desktop Publishing. Electronic Learning, 7 (6), pp. 223-225.

McGinty, R.S. and Sautter C. (1987, January) Software Trends and Trend Setters. Electronic Learning, 7 (1), pp. 18-19.

McKelvie, S. (1987, June) Sex Differences, Lateral Reverse and Prose as Factors in recognition Memory for Photographs of Faces. The Journal of General Psychology, 114 (1), pp. 13-37.

McKim, A.H. (1972) Experiences in Visual Thinking. Montery, CA: Brooks/Cole.

Metallenos, N. and Chartland, S. (1987) Comprehension and Recall of Television's Computerized Images: An Exploratory Study. Precedings of Selected Research Paper Presentations at the 1987 Annual Convention of the Association for Educational Communications and Technology: Atlanta, GA.

Miller, S.K. (1987) Selecting and Implementing Educational Software. Boston: Allyn and Bacon, Inc.

Moore, M.V. and Nawrocki, L.H. (1978) The Educational Effectiveness of graphic Displays for Computer Assisted Instruction. Alexandria, VA: Army Research Institute for the Behavioral and Social Sciences. (ERIC Document Reproduction Service No. ED. 169197).

Morrison, G.R., Ross, S.W., O'Dell, J.K. and Shultz, C.W. (1988) Screen Density and Text Density: Getting More out of Less with CBI. Proceedings of Selected Research Paper Presentation at the 1988 Annual Convention of the Association for Educational Communications and Technology: Atlanta, GA.

Mouly, G.J. (1973) Psychology for Effective Teaching. New York: Holt, Rinehart and Winston.

Mulkeen, T.A. (1987) Proceedings of Selected Research Paper Presentations at the 1987 Annual Convention of the Association for Educational Communications and Technology, Atlanta, GA.

Murch, G.M. (1973) Visual and Auditory Perception. Indianapolis: Bob-Merril.

Naiman, A.H. (1987) A Hard Look at Educational Software. Byte, 12 (3), pp. 193-200.

National Education Association (1988) A Teacher Survey NEA Report: Computers in the Classroom. Washington, D.C.

Nelson, R.P. (1981) The Design of Advertising (4th ed.). Dubuque, Iowa: Wm. C. Brown Company Publishers.

Nesbit, L.L. (1987) Relationship Between Eye Movement, Learning and Picture Complexity. In F.H. Dwyer (Ed.), Enhancing Visualized Instruction: Recommendations. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher.

Norman, D.A. (1987) Design Principals for Human Computer Interfaces. In R. Gagne (Ed.), Instructional Technology: Foundations. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher.

Nunan, T. (1984) Countering Educational Design. London: Croom Helm.

Orihuela, G. (1986) Guidelines for Teachers and Instructional Designers for the Preparation of Instructional Text Material and Instructional Graphic Materials Using a Microcomputer. Ann Arbor, MI: U-M-I Dissertation Information Service.

Orlando, L.C. (1988) NSBA Issues Bold Statement to Washington About Technology. Electronic Learning, 7 (6), p. 8.

Orna, L. (1983) Writing to Form. Kent Br7 5sn: Ravensbourne College of Art and Design.

Otte, R.B. (1984, October) Courseware for the 80's. T.H.E. Journal, 12, pp. 89-91.

Patton, H. (1988, January) Modern Masterpieces. A V Communication, 22 (1), pp. 26-29.

Pexdek, K. (1987) Television Comprehension as an Example of Applied Research in Cognitive Psychology. In D. Berger, K. Pexdek and W.P. Banks (Eds.), Applications of Cognitive Psychology: Problem Solving, Education and Computing. Hillsdale, NJ: Lawrence Earlbaum Associates, Publisher.

Phillips, M.G. (1981) The Education/Training of a Trainer. NSPI Journal, 20 (1), pp. 14-20.

Pliske, D.B. and Psotka, J. (1986, Spring) Exploratory Programming Environments for Designing ICAI. Journal of Computer-Based Instruction, 13, pp. 52-57.

Reed, S. and Sautter, R.C. (1987, May/June) Visions of the 1990's: What Experts Predict for Educational Technology in the Next Decade. Electronic Learning, 7 (5), pp. 18-23.

Rigney, J.W. and Lutz, K.A. (1976) Effects of Graphic Analogies on Concepts in Chemistry and Attitude. Journal of Educational Psychology, 68, pp. 305-311.

Roblyer, M.D. (1983, Fall/Winter) Toward More Effective Microcomputer Courseware Through Application of Systematic Instructional Design Materials. AEDS Journal, 17, pp. 23-32.

Roth, S. (1987) Educational Criteria Used by manufacturers in the Development and Publication of Educational Software. Unpublished Doctoral Dissertation, Oklahoma State University, Stillwater, OK.

Ruckerman, R.A. (1983) Computers and Teacher Training. AEDS Journal, 16 (2), pp. 123-130.

Salomon, G. (1985, November) Information Technologies: What You See Is Not Always What You Get. Educational Psychologist, 20 (10), pp. 207-210.

Samuels, S.J. (1970) Effects of Pictures on Learning to Read, Comprehension and Attitude. Review of Educational Research, 40, pp. 397-407.

Satterthwaite, L. (1984) Guidelines for Effective Visuals. Instructional Innovator, 29 (5), pp. 16-17.

Salvay, M.L. (1988, March) Sunburst's Marge Kosel: Striving to Set the Software Standard. Electronic Learning, 8 (6), pp. 924-927.

Simon, M. (1988, March) Let's Help Teachers Harness Computer Power. Electronic Learning, 7 (6), pp. 6-8.

Simmons, G.P. (1987) An Analytical Study of Three Design Principals Used in the Development of Instructional Materials. Unpublished doctoral dissertation, Indiana University, Indiana.

Staff (1987, September) Computer Teaching Lagging in Schools. The Sunday Oklahoman, p. 20.

Steffin, S.A. (1983, January) A Suggested Model for Establishing the Validity of Computer Assisted Instructional Materials. Educational Technology, 23, pp. 17-19.

Sutherland, S. and Winn, W. (1987) The Effect of the Number and Nature of Features and of General Ability on Simultaneous Successive Processing of Maps. Proceedings of Select Paper Presentations at the 1987 Annual Convention of the Association for Educational Communications and Technology, Atlanta, GA.

Taylor, R.P. (1980) The Computer is the School: Tutor, Tool and Tutee. New York: Teacher' College Press.

Tennyson, R. and Park, O. (1986, October) Computer-Based Response Sensitive Strategies for Selecting Presentation Form and Sequence of Examples in Learning of Coordinate Concepts. Journal of Educational Psychology, 78 (2), pp. 153-158.

Tatebaum, T.J. and Muliken, T.A. (1986, Winter) Computers as an Agent of Change. Computers in Education, 2, pp. 91-103.

Treisman, A.M. (1974) Selective Attention in Man. In P.A. Fraied (Ed.), Readings in Perception: Principal and Practice. Lexington, MA: Heath.

Trollop, S.R. (1987, May) Four Computer Features to Avoid. Computing Teacher, 14 (8), pp. 33-35.

Tullis, T.S. (1983) The Formatting of Alphanumeric Displays: A Review and an Analysis. Human Factors, 25, pp. 657-682.

Tayler, J.G. (1983, February) Your Prescription for CAI Success. Instructional Innovator, 28, pp. 25-26, 40.

Van Bergeijk, W.A., Pierce, J.R. and David, E.E. (1960) Waves and the Ear. Annual Review of Psychology, 32, pp. 21-52.

Wallington, P. (1984) Report of Educational Uses of Computers in New Jersey School Districts. Trenton, NJ: State Department of Education. (ERIC Document Reproduction Service No. ED. MF01/PC06).

Waugh, G. and Carrier, D. (1986, Spring) Computer-Based Education What We Know and Need to Know. Journal of Computers in Mathematics and Science Teaching, 5 (3), pp. 13-15.

Wallington, C.J. (1981) Generic Skills of an Instructional Developer. Journal of Instructional Development, 4 (3), pp. 28-32.

WGBH (Producer) (1988) Top Gun and Beyond. (Videotape) Boston, MA: Public Broadcasting System.

White, M.A. (1988, January) The Third Learning Revolution. Electronic Learning, 8 (4), pp. 6-7.

Williams, C. (1987) Making a Hit: Here's What the Developers Say. Electronic Learning, 8 (4), pp. 20-23.

Williams, F. And Williams, V. (1985) Success with Educational Software. New York: Praeger Publishers.

Willis, F.H. (1965) Fundamentals of Layout. New York: Dover Publications.

Winn, B. (1986, April) Simultaneous and Successive Processing of Circuit Diagrams Having Different Amounts of Detail. Proceedings of Selected Research Paper Presentations at the 1986 Annual Convention of the Association for Educational Communications and Technology, San Francisco, CA.

Winn, B. (1987) Real-Time Graphics for CAI: A Rudimentary Grammer and Demonstration Program. Proceedings of Selected Research Paper Presentations at the 1987 Annual Convention of the Association for Educational Communications and Technology, Atlanta, GA.

Winn, A. and Tidhar, C.E. (1988), January) Learning to View - Viewing to Learn. TECH Trends, 33 (1), pp. 12-16.

Ullmer, E.S. (1988) On the Form of Human Enterprise: Examining the Design of Design. Educational Technology, 23 (9), pp. 4-21.

Yeaman, A.R. (1987) Electronic Text Display: An Experiment in Electronic Page Turning and Window Effect. Proceedings of Selected Research Paper Presentations at the 1987 Annual Convention of the Association for Educational Communications and Technology, Atlanta, GA.

Zuckerman, R.A. (1988) Computers and Teacher Training. AEDS Journal, 16 (2), pp. 123-130.

APPENDIXES

APPENDIX A

RESEARCH INFORMATION FOR THE SURVEY

"DISPLAYS AND COMMUNICATION" by Indiana University Professor Emeritus Malcolm Fleming. This Chapter 9 in Robert Gagne's (Ed.) 1987 book, Instructional Technology: Foundations. Hillsdale, NJ: Lawrence Earlbaum Associates, Publishers.

1. Attention is highly selective (Treisman, 1974). We can give attention to only a small part of the environment at one time, and of that we see most sharply only the tiny, central portion of the visual field.

2. Attention is drawn to what is novel or different (Berlyne, 1970). By manipulating instructional displays the designer can readily introduce novelty. For example, in speech change the volume or inflection; in print, change the typeface or color; in film or television, change the pace or introduce a novel element such as a sound effect.

2a. Changes in brightness and particularly in motion are strong attention-getting factors.

2b. Once attention is gained, continuing changes in the ongoing stream of instruction can help maintain it.

3. Attention is drawn to moderate complexity (Forgus, 1966). Obviously, this can be overdone, leading to learner avoidance. On the other hand, a too simplistic display may get very little attention.

4. Lean displays focus attention. This has been called attribute isolation (Bovy, 1981). The procedure here is simply to include only the most relevant information.

5. Learned cues can direct attention (Bovy, 1981). Examples are arrows, underlinings, circles, or rectangles around items. Such attention-directing cures are effective only with "literate" learners. Another very effective cue is simply to direct the learner verbally to look for or listen to certain features. Captions can have a strong effect on the amount and kind of attention given pictures.

7. Moderate uncertainty may induce careful attention (Mouly, 1973). This implies that displays that are too easy or obviously may fail to gain or hold attention. The amount of uncertainty may vary with the task, less for concepts and more for problem solving.

8. Perception is organized (Eysenck, 1984). Learners try to construct meaningful wholes from the environment: objects events, ideas.

8a. Perhaps the most basic organizational step in the perceptual process is the separation of the visual field into figure and ground. For example, key figures (objects, persons) in a picture are selected and given more attention than the background scene. The designer should make the essential information figural and therefore dominant.

9. Perception is relative (Helson, 1974). Perception is not registered in absolute values: rather, it functions by comparison.

9b. Judgements of size or quantity are relative. The size of an unknown pictured object cannot be determined without reference to some standard, for example a hand or ruler.

9c. Judgements of depth or distance are relative. For instance, size and depth are inversely related. Perceived size in a picture or screen is influenced by the frame - large if it is filled and smaller if it's not.

9e. Judgements of motion are relative. Perceived motion is relative to some reference point, for example, the frame or background in a picture.

11. Vision is most sensitive to colors in the middle of the spectrum, yellow and yellow-green, and least sensitive to those at the ends of the spectrum, violet/blue and red. Similarly, audition is most sensitive to pitches in the middle, two to three octaves above middle C, falling off toward both lower and higher pitches (Murch, 1973; Van Bergeijk, Pierce, & David, 1960).

APPENDIX B

PARTICIPANTS IN THE SURVEY

The 1988 Educational Software Preview Guide's
Recommended Publishers

American Guidance Service, Inc.

Apple Computer Co.

Ashton-Tate

Atari Corp.

Automated Language Processing Systems

Baudville'Beagle Brothers

Bedford Software Ltd.

Blue Lion Software

Borland International

Broderland Software

C & C Software

Cactus

Cambridge Development Laboratory

Challenger Software Corp.

Chancery Software LTD.

Claris Corp.

Commodore Computer Systems Div.

COMPress

Compu-Teach

Conduit-University of Iowa

Creative Publications

Cricket Software

D. C. Health & Co.

Davidson & Associates, Inc.

Decision Development Corp.

Design Science

Designware, Inc.

DLM

Earthware Computer Service

Ed. TEch. Center/Harvard Grad. Schl. Ed.

Educational Activities, Inc.

Edusoft

Edtech

Electronic Arts

Epyx, Inc.

Exsym

Focus media

Freesoft Co.

Gamco Industries

Gessler Educational Software

Great Wave Software

Grolier Electronic Publishing, Inc.

Harcourt Brace Jonanovich Hartley Courseware

Hayes Microcomputer Products, Inc.

Holt, Rinehart and Winston

Houghton Mifflin Co.

HRM Software/A division of Quenue, Inc.

Humanities Software IBM Educational Systems

Innovision

Island Software

John Wiley & Sons, Inc.

Jostens Educational Technology

Krell Software Corp.

Learning Technologies

Lego Systems, Inc.

Letraset USA

Living Videotext, Inc.

Logo Computer Systems

Lotus Development Corp.

Mark Davis

Marshware

McGRaw-Hill Book Co./School Div.

MECC

Meizner Bussin Machines Inc.

Mentor Learning Systems Inc.

Microsoft Corp.

Midwest Publications Co., Inc.

Miliken Publishing Co.

Mindplay, Inc.

Mindscape, Inc.

Nashoba Systems Inc.

National Geographical Society

Newsweek Inc.

Optimum Resources, Inc.

Paperback Software

Passport Designs, Inc.

Prentice-Hall Allyn and Bacon

PTI - Koala

Rand McNally & Co.

Random House School Division

Savtek Corp.

Scholastic, Inc.

Scott Foresman and Co.

Sensible Software

Shenandoan Software

Silicon Beach Software, Inc.

Silver Burdett & Ginn

Simon & Shuster Software

Society for Visual Education

Softswap

South-Western Publishing Co.

Spinnaker Software, Inc.

Springboard Software

Styleware, Inc.

Sunburst Communications

Teach Yourself by Computer Software, Inc.

Technical Educational Consultants

Temporal Acuity Products, Inc.

Terrapin, Inc.

Time Education Center

Tom Snyder Productions True Basic, Inc.

Venier Software

Word Perfect Corporation

APPENDIX C

INITIAL PARTICIPANT CONTACT LETTER

OSU

Oklahoma State University

CLEARINGHOUSE OF INFORMATION
ON MICROCOMPUTERS IN EDUCATION

STILLWATER, OKLAHOMA 74078
108 GUNDERSEN
(405) 624-6254

10 July, 1988

Dear Sir or Madam:

Your company was identified by the 1988 Educational Software Preview guide, developed by the Educational Software Evaluation Consortium, as publishing software of high quality. I am conducting a study of design procedures and criteria used by software publishers and/or their software developers in the design and layout of the software they publish. The results of this study will by published in the CHIME Newsletter.

I ask that you please take a few minutes to fill out the enclosed survey and return it as soon as possible to our office. At no time will individual publishers be identified by software or name with respect to the information you return via the survey. Cooperating publishers, however, will be listed as participants in the study.

If you have any questions regarding the above request, please contact me at my office, 405-624-7124. Your cooperation is appreciated by CHIME and by me.

Sincerely,

David Payne

Carl David Payne Jr.
Project Coordinator

CENTENNIAL
DECADE
1980 • 1990

APPENDIX D

THE SURVEY

The Clearing House for Information on Microcomputers in Education (CHIME), at Oklahoma State University, is committed to ongoing research regarding all elements of educational computing. We would greatly appreciate your assistance in taking a few minutes of your time to complete the front and back of this survey and return it in the enclosed envelope.

1. For software that your company has determined to publish, who has the creative responsibility for the development of the visual design and /or layout? (check only one)
___ free-lance writers ____ company designers ___ the publisher
Comments:

2. What is the educational level of the person with the creative responsibility for the development of the visual design and /or layout of your software? (check degree & list major subject area)
HIGHSCHOOL ______________________ M.A./M.S. ______________________
B.A./B.S. ______________________ Ed.D./PhD. ______________________

3. Number of courses or hours of training in: (respond to area(s) that apply to you)

______ COMPUTER PROGRAMING
(study of languages, programing, retrieval and storage systems, etc.)
_____ EDUCATIONAL PSYCHOLOGY
(study of the nature and operation of human learning)
______ INSTRUCTIONAL DESIGN
(study of systematic designs for instruction, educational materials, and technology)
_____ OTHER
______ EDUCATIONAL COMPUTING
(study of computing in the design, analysis and presentation of instruction)
______ GRAPHIC DESIGN
(study of the art/science of presenting the printed word or illustrations)
______ VISUAL DESIGN
(study and design of materials that appeal directly to sight, i.e. TV, film, advertising)
______ UNKNOWN

4. Number of years teaching experience of the person with creative responsibility at:
(respond to those areas apply to you)
_____ ELEMENTARY LEVEL _____ SECONDARY LEVEL _____CITY/COUNTY VO-TECH
_____ JUNIOR/COMMUNITY COLLEGE LEVEL ______ UNIVERSITY/COLLEGE LEVEL
_____ OTHER ______ UNKNOWN

ATTENTION

Attention , as defined and used here, is the act and/or ability to keep one's mind closely on something. Without attention there can be no learning. Designers should seek to obtain the learners attention and to keep it.

1. Do your software designers create software that places the most pertinent information:
(Check one)
____ evenly over the entire visual field ____ over the upper one-third of the visual field?
____ over the central portion of the field? ____ over the bottom one-third-third of the visual field?

2. Do your software designers attempt to design visual displays that are: (Check one)
_____ of simple complexity ? _____ of moderate complexity ? _____ of extreme complexity?

3. Do your software designers attempt to focus the users attention through visual displays that are: (Check one)
______ lean - includes only the most relevant information ?
______ moderate - includes relevant information and a degree of realism.
______ realistic - attempts to approximate reality

4. Do your software designers manipulate instructional displays by: (one check for each item please)

	yes	no	occasionally	comments
a. changing the volume or inflection of speech?				

	yes	no	occasionally	comments
b. changing the color or type face of your print?				
c. changing the pace of the action or instruction?				
d. changing the rate of the motion of the visual field?				
e. changing the brightness of the visual field?				
f. inducing some change in the ongoing stream of instruction once attention has been gained?				

5. Do your software designers attempt to guide attention by using the following cues?

	yes	no	occasionally	comments
ARROWS				
UNDERLININGS				
CIRCLES				
RECTANGLE ABOUT ITEMS				
INVERSE LETTERING				
VERBAL DIRECTIONS				
CAPTIONS FOR VISUALS				
OTHER				

6. Software designers often attempt to induce careful attention by manipulating the degree of uncertainty in the visual or display. Do your designers:

A. when working with concept questions attempt to achieve:

____ no uncertainty in the program?
____ moderate uncertainty in the program?
____ a lot of uncertainty in the program?

B. when working with problem solving questions attempt to achieve:

____ no uncertainty in the program?
____ moderate uncertainty in the program?
____ a lot of uncertainty in the program?

PERCEPTION

7. Perception, as defined here means to grasp mentally, or to become aware of through sight, hearing, or touch. With this in mind, please responds to the following. Do your software designers:

	yes	no	occasionally	comments
a. Separate the visual field into figure and horizon				
b. Give more attention to key figures, object s or persons than the background scene?				
c. Use a recognizable standard to help determine size or quantity of an unfamiliar object?				
d. Use a reference point, such as the frame or background, to judge the rate and or direction of motion?				
e. Use colors that are primarily in the middle of the color spectrum throughout the program?				
f. Use sound pitches that are two to three octaves above or below Middle C? (normal?)				

8. Who, in your company, is responsible for the final visual design and/or layout of the final published version of the software that your company decides to market? (choose one)

____ the software publisher? ____ the software designer ? ____ the software programer?

9. Education, major, and years of experience of the person responsible for the final creative design /layout of the software programs that you publish?

APPENDIX E

THE SECOND PARTICIPANT LETTER

OSU

Oklahoma State University

CLEARINGHOUSE OF INFORMATION
ON MICROCOMPUTERS IN EDUCATION

203 Gundersen Hall
Stillwater, OK 74078
(405) 624-7124

Dear Sir or Madam:

Please return the survey on attention and perception that was recently sent to you. The survey represents the latest information compiled from current research. This information is to be shared with and will be of great interest to the more than 500 members of the Clearing House for Information on Microcomputers in Education (CHIME), mostly microcomputer teachers. Our reviews and articles have helped many of these educators make informed choices when it comes time for software purchases. Your participation helps us all.

David Payne
Project Coordinator

CENTENNIAL
DECADE
1980 • 1990

2

VITA

Carl David Payne Jr.

Candidate for the Degree of Doctor

of Education

Thesis: DESIGN STANDARDS AND CRITERIA FOR GRAPHICS USED BY PUBLISHERS AND MANUFACTURERS IN THE DEVELOPMENT OF EDUCATIONAL SOFTWARE

Major Field: Curriculum and Instruction

Biographical:

Personal Data: Born in Tulsa, Oklahoma, March 9, 1948, the son of Carl D. and Helen E. Payne. Married to Laurie M. Scales-Payne on July 15, 1978.

Education: Graduated from Nathan Hale High School, Tulsa, Oklahoma in May, 1967; received Bachelor of Science degree from Oklahoma State University, Stillwater, Oklahoma, May 1971; received Master of Science degree from Oklahoma State University, Stillwater, Oklahoma, December, 1972; completed requirements for the Doctor of Education degree at Oklahoma State University, Stillwater, Oklahoma May, 1989.

Professional Experience: Classroom Teacher, Montclair College Preparatory School, Van Nuys, California, September, 1981 to June 1982; Classroom Teacher, California Preparatory School Van Nuys, California, September 1982, to June 1983; Classroom Teacher, Cathedral High School, September, 1983 to January, 1985; Classroom Teacher, Chatsworth High School, Chatsworth, California, January, 1985 to June 1985; Graduate Assistant, Oklahoma State University, Stillwater, Oklahoma, September, 1985 to June 1987; Instructor, Oklahoma State University, Stillwater, Oklahoma, June, 1987 to August, 1988; Teaching Assistant, Stillwater, Oklahoma, August, 1988 to present; Adjunct Faculty Member, Central State University, Edmond, Oklahoma, August 1988 to present.