DOCUMENT RESUME

ED 319 374 IR 014 392

AUTHOR	Char, Cynthia
TITLE	Research and Design Issues Concerning the Development of Educational Software for Children. Technical Report No. 14.
INSTITUTION	Bank Street Coll. of Education, New York, NY. Center for Children and Technology.
SPONS AGENCY	National Inst. of Education (ED), Washington, DC.
PUB DATE	Apr 83
CONTRACT	300-81-9375
NOTE	7p.; Paper presented at the Annual Meeting of the American Educational Research Association (Montreal, Canada, April 11-14, 1983).
PUB TYPE	Reports - Research/Technical (143) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Computer Games; *Computer Science Education; Computer Simulation; *Computer Software Development; Computer Software Evaluation; *Courseware; Data Collection; Elementary Education; Field Tests; *Instructional Design; Programing; Teacher Attitudes; Teacher Role

ABSTRACT

Several research and design issues to be considered when creating educational software were identified by a field test evaluation of three types of innovative software created at Bank Street College: (1) Probe, software for measuring and graphing temperature data; (2) Rescue Mission, a navigation game that illustrates the computer's use for simulation; and (3) Whale Search and Treasure Hunt, games that introduce the notion of programming to children. The field test used a case-study approach to examine the comprehensibility, appeal, and usability of the programs. The most striking finding was the range of use of the software in different classrooms, including differences in the proportion of students having access to the computer, the amount of time each student used the software, the degree and type of teacher involvement, and classroom organization. It was also found that the use of software was influenced by computer and teacher resources, as well as by prior teacher training in and perceptions of science, math, or computers. Software design implications point to the need for creating software that allows students to work in a collaborative fashion independent of the teacher and for addressing comprehensibility at all levels. It is recommended that teachers be provided with a conceptual framework for software. (MES)

Research and Design Issues Concerning the Development of Educational Software for Children

Cynthia A. Char

Technical Report No. 14

RESEARCH AND DESIGN ISSUES CONCERNING THE DEVELOPMENT OF EDUCATIONAL SOFTWARE FOR CHILDREN*,**

Cynthia A. Char

Center for Children and Technology
Bank Street College of Education

I would like to discuss a number of different research and design issues that need to be considered when creating educational software for classroom use by children. My comments stem from a fieldtest evaluation that was conducted on three types of innovative software created at Bank Street College, which were produced as part of a multimedia curriculum package on science and mathematics for fourth through sixth graders.

At the Project in Science and Mathematics Education, we are producing a television series, microcomputer software, print materials and, eventually, videodiscs that provide an integrated approach to science and math instruction. At the core of the Project is a 26-episode dramatic series for television, called "The Voyage of the Mimi." The series follows the adventures of two young scientists and their teen-aged crew who are studying whales while aboard a research vessel. Like other media components of the Project, the series is designed to provide students with an appealing and compelling view of what it is like to do science and to be scientists, and how mathematics and technology can be used in scientific inquiry.

In order to engage children in some of the ways computers are used in the world by adults, three different types of software have been developed to accompany the television series. One software piece, Probe, displays the computer's usefulness in data collection and representation and, at present, is a software package for measuring and graphing temperature data. Eventually, the software will be able to gather data on light and sound as well. The second piece of soft-

---

*Paper presented at symposium of the American Educational Research Association, "Chameleon in the Classroom: Developing Roles for Computers," Montreal, Canada, April 1983.

**The research reported in this paper was funded by the Department of Education, Contract # DE-300-81-0375.

ware, Rescue Mission, is a navigation game that illustrates the computer's use for simulation. It motivates students to apply such skills as map reading and mathematical knowledge of scale, degrees, and angles to the real-world problem of ocean navigation. The third type of software consists of two games, Whale Search and Treasure Hunt, which introduce the notion of programming to children. To play these games, children must learn the basic commands of the Logo language to move their "ship" about the computer screen, either to reach a trapped, netted whale or some hidden treasure. Given the limited computer resources that exist in most schools, all four software pieces were designed for use by more than one student at any given time; for example, the navigation simulation game can accommodate up to 12 student players at a time.

Evaluating software designed for school use, as opposed to home use, had certain implications for the way we went about studying the effectiveness of the software. First, we were interested in seeing how the materials were used in natural classroom groupings and settings, rather than by individual or small groups of children outside the classroom. Second, we placed a special focus on teachers and their role in the classroom, rather than an emphasis only on children and their reactions to the software. We were particularly interested in seeing how teachers viewed our materials, made decisions about how to use them, and evaluated the software experience.

In order to take an intensive look at how computers are used in classrooms, and how software use is affected by teachers' views and roles, we conducted a fieldtest using a case-study approach. Thirteen classrooms, drawn from seven schools in the New York metropolitan area, participated in the fieldtest. All the classrooms were in the Project's targeted age group of fourth through sixth grade, and were diverse with respect to ethnic, class, and urban/suburban variables. The fieldtest was unstructured so as to give teachers flexibility in using the materials, and in their selection and organization of classroom lessons.

To evaluate the effectiveness of our software, three different aspects of the materials needed to be examined: comprehensibility, appeal, and usability. Furthermore, we wished to address each of these aspects from the perspectives of teachers, students, and our own staff of researchers. For example, in order to obtain a collective picture of student comprehension of a piece of software, teachers were asked what aspects seemed unclear to students; students were asked what questions they had about the materials; and students were administered a written test focusing on specific software tasks which we, as researchers, felt might be problematical for children.

Seven different types of measures were used in the fieldtest. These included classroom observations, student and teacher interviews, written forms where teachers described and evaluated the Project activities used each day, log-in books where students recorded their names and time spent at the computer, evaluation forms where students indicated their reactions regarding the software's appeal and comprehensibility, and teacher background information forms where teachers described their past teaching experiences and the science and mathematics instruction in their current classrooms. There were three different versions of each measure, one version for each type of software.

The most striking finding of the fieldtest was the considerable range of use of the software in different classrooms. Some of these differences were quantitative, such as the proportion of students in a class having access to the computer, and the amount of time each student used the software. Other differences were qualitative, such as the degree and type of teacher involvement in the software experience, and the ways teachers organized their classrooms to allow some children to work on the computer while others were engaged in noncomputer activities. For example, for the temperature gathering and graphing software, some teachers assumed the role of demonstrator, thereby engaging the whole class but limiting the amount of hands-on time students had with the computer. Other teachers chose a less central role and acted as resource persons for students working independently on temperature experiments at the computer. Still others acted as "software managers" and mainly ensured that the students had the necessary materials, that the software was working properly, and that students were taking turns in a relatively fair manner.

The amount and the <u>way</u> software was used in classrooms appeared to be greatly influenced by two factors. The first factor involved computer and teacher resources, and the ratio between students and computers and between students and teachers. In classes with more than one computer or with regular access to a computer, proportionately more students were able to use the software for longer periods of time, and were able to take greater advantage of the learning experiences afforded by the software. Similarly, in classrooms where there were only 15 to 20 students, as opposed to 35, teachers were able to take a more active role in the software experience--not as demonstrators, but as active participants who could facilitate and monitor children's progress with the software.

The second factor involved teachers' prior training in and perceptions of science, mathematics, or computers. For example, some of the teachers who used the temperature graphing software were those who had had previous training in science, and who taught science as a sequential and coherent curriculum, emphasizing experimentation. These teachers tended to find the software very useful, recognized its many applications, and allowed children to work as teams of scientists and rotate through the computer station to gather data for their temperature experiments. In contrast, other teachers participating in the fieldtest had little formal training in science. They taught in schools with little emphasis on science instruction, and usually taught science as a series of lessons on various topics such as plants or animals. These teachers also liked the temperature software, but had difficulty seeing it as a flexible data-gathering and graphing tool; they were less able to generate activities and experimental contexts for the software beyond those provided in the manual. They also seemed to be less able to follow through with student questions about heat and temperature, or to notice when children were having conceptual problems with the software.

Interestingly, computer experience per se on the part of teachers did not guarantee software "success." One of the teachers who used the navigation simulation was a math teacher with prior experience in teaching children computer programming, but didn't know quite what to make of the software. He chose not to become actively involved in monitoring or facilitating students' progress with the simulation, having viewed it as an experiential learning activity with a self-sufficient and self-contained context. He also felt that the software was designed to teach navigation, rather than map-reading skills and math concepts such as angles, degrees, measurement, and triangulation. Stated another way, he felt that navigation was the curriculum content, rather than the vehicle through which to teach various math skills. Thus, our simulation and temperature gathering software may fall in a curious spot somewhere between structured CAI and computer programming, and may not automatically rest in a conceptual "niche" of computer experiences commonly found in schools.

What are the software design implications of these findings? First, to address the scarce computer resources and less-than-optimal student-teacher ratios found in most schools, it is important to begin creating software which can be used by more than one child at a time, and which allows students to work in a collaborative fashion independent of the teacher. To ensure effective use of the software, its comprehensibility to students needs to be addressed at a variety of levels --from their understanding of the text and graphics presented on each screen, to their understanding of the software's general objective, the purpose of each menu item, and how to "get around" the program. Comprehension can be facilitated by paying careful attention to the way specific screens are designed, programs are structured, and student/teacher manuals are written, with obvious advantages inherent in software and manuals which incorporate various skill and conceptual levels.

If software is designed to meet this range of student needs, teachers will be better able to work with students at a higher level of learning strategies and outcomes, and to help them make connections to other classroom activities, rather than having to explain specific text on the screen or what key to press. Teachers will also be freer to work with students engaged in noncomputer activities. Furthermore, such software will allow students to help each other and to solve problems by themselves, fostering collaborative work, as well as possibly speeding up students' time at the keyboard, thus providing a greater number of student turns at the computer.

To address the needs of teachers with limited training in or narrower views of science, mathematics, and computers, one should provide teachers with a conceptual framework for the software. This framework should describe the software's approach, educational objectives, and specific math and science skills and concepts, as well as outline connections to other activities and materials in science, math, and other subject areas. This could be done in a teacher manual or guide, or as part of a videotape or written presentation for teacher training sessions. Background material on various math and science content areas or references should also be available in the teacher's guide.

These design implications are currently being incorporated into the development of the Project software, which will undergo revision before being subjected to yet another round of formative testing in schools. Also under way are plans for the Project's teacher training component, which will address the importance of teachers and their role in classrooms when trying to implement innovative uses of educational software. With educational software and the presence of computers in schools still in their infancy, it is difficult for software developers to utilize the computer's graphic and interactive capacities and take students along new and exciting educational avenues, while still being sensitive and accountable to the needs and interests of teachers. The Project in Science and Mathematics Education, with its staff of producers, curriculum specialists, and researchers, models an attempt to bridge the worlds of the software designer, the educational visionary, and the classroom teacher. It is our view that we, along with others, should continue this pursuit to create innovative educational software that can be used in classrooms today, while exploring new directions for software to be used in the classrooms of the future.