DOCUMENT RESUME

ED 237 082 IR 020 033

AUTHOR	Zeitz, Leigh
TITLE	Writing Software for Education.
PUB DATE	Jun 83
NOTE	4p.; Reprinted from "Personal Computer Age," March 1983.
PUB TYPE	Journal Articles (080) -- Guides - Non-Classroom Use (055)
JOURNAL CIT	Computer-Using Educators Newsletter; v5 n6 p11-12 June 1983
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Programs; Elementary Secondary Education; Guidelines; *Instructional Development; *Material Development; Teacher Developed Materials
IDENTIFIERS	PF Project

ABSTRACT

The guidelines in this article help in developing educational software. The first rule is that educators should not develop software for education, but should develop an "educational system" that uses software. The distinction is that the former is primarily concerned with the computer and its requirements while the latter focuses on educating the student. The steps involved in developing a system are: (1) specify goals; (2) specify objectives; (3) decide on an instructional method; (4) analyze the audience; (5) evaluate the continuum; and (6) write the system and any computer programs it may require. An educator does not need to be a programmer; professionals can translate the description into a program. The final step is to field test the system and make any necessary changes. (DC)

WRITING SOFTWARE FOR EDUCATION

by Leigh Zeitz

Article appearing in Computer Using Educators Newsletter, vol. 5, no. 6, June 1983. Reprinted with permission from the newsletter of Computer-Using Educators, Box 18547, San Jose, CA 95158.

Writing Software for Education

Taken from March, 1983 issue of Personal Computer Age, by Leigh Zeitz.

Many of you have shown an interest in doing more than teaching with commputers in your single schools or school districts. You have been contacting me with questions about how you might get in on writing software for education. I thought it might be appropriate to go over some guidelines in developing software for education.

When it comes to developing software for education I have only one rule for you:

DON'T

I repeat: Do not develop software for education!

Contrary to popular belief, computer software does not educate.

An educator educates and computer software is only a medium of presentation for an instructional system that an educator has developed.

There is a distinct difference between "writing software for education" and "writing educational systems* that are to be used on a computer."

* I will use the phrase "educational system" instead of "educational program," which may be more accurate, in an attempt to reduce any confusion with the term "computer program."

When you're <u>writing software for education</u>, the project is viewed from the keyboard of a computer. It's thought of as a computer task to be used in education rather than an educational system that's being used on a computer. The developer is usually more concerned with which language to use and how to best conserve valuable memory space than whether the instructional design is valid.

Conversely, when you're writing <u>educational systems to be used on a computer</u>, the project is viewed from the educational setting. It's thought of as a full educational system that is created according to the guidelines of effective instructional design. This attitude allows the designer to admit that this system is not limited to the computer in its methods of presentation. Along with the computer, this system may (and should) integrate printed material, classroom discussion, video tapes, and projects that cause the students to apply their newly acquired information to the real world.

DEVELOPING YOUR EDUCATIONAL SYSTEM

With this distinction in mind, let's begin to consider the steps you should follow when developing an educational system.

SPECIFY YOUR GOALS

Before you do anything, you must give yourself direction by setting some goals for your project. What sort of skills achievement do you expect to see from the students who work through your system?

Usually, your goals will fall into one of three categories: 1) Introduction of new material, 2) Remediation of previously taught material, or 3) Enrichment and application of previously learned material. Each of these will require a different form for presentation of the material and evaluation. So, it's necessary to know what you're hoping to achieve.

An example of a goal might be "This system will introduce a student to the use of prefixes and suffixes."

SPECIFY YOUR OBJECTIVES

Goals give you direction and objectives give you targets.

Objectives are more specific than goals. They state exactly what you're trying to achieve. They're written using terms that describe changes in the student's behavior that are observable by others. This way the success of the project can be evaluated in terms of what the students can do rather than what someone thinks he knows.

"The student will know five prefixes." This is an objective that can't be measured. How can anyone measure whether the student knows the prefixes or not?

"The student will correctly use five prefixes." This objective can be measured. It's a simple task for someone to decide if five prefixes have been used and if they've been used correctly.

EVALUATE THE MEDIA

Now that you know what you want to do, you must decide how you'll go about doing it. What's the best way to implement your educational system?

You may want to center the activities around the computer as a delivery device, but is that the best way to do it? There's a place for all forms of computer-based education including drill and practice, tutoring and simulations. But, remember to ask yourself whether the most effective means of presenting the material is through computer or the traditional ways, or should it be a combination of both?

If you're developing this for yourself, make sure that you consider the number of computers that you'll have at your disposal. It would be senseless to use the computer for a lot of drill and practice when you only have access to a couple of terminals which would be better utilized as tutors or simulation devices.

continued on page 12

continued from page 11

ANALYZE YOUR AUDIENCE

Now, before you actually start writing, you must determine who it is that will be using your system. Will it by used in the home or the school market? Do you intend for your pupils to be at the preschool, elementary, secondary, or adult levels? What will be the reading level necessary for a student to succeed in your program?

This is the part that the educational publishing industry fondly refers to as the "Scope and Sequence." The Scope is an analysis of your goals, objectives, and intended audience. You've already done that part. The Sequence is an outline of how you'll teach the objectives you've set for yourself. It's a logical sequence of steps for presenting your material on the proper gradient of increasing difficulty, and with the proper amount of practice and evaluation.

EVALUATE YOUR PROGRAM

Before you go to all the effort of programming your lessons, check to see if your continuum makes sense. This is where you should enlist the services of a resource specialist in your school or community. These people are specially trained in what works in teaching their given subject and what doesn't. Too often we can get so close to a product that we can't see holes in our academic logic that are big enough to drive a truck through.

Get a second opinion!

WRITE YOUR EDUCATIONAL SYSTEM

Now, it's time to write your educational system and any computer programs that you wish to include. I'll bet that you didn't realize how much work you would have to do before you got down to the actual writing.

I would like to make a point here about writing your computer program: <u>You don't have to do it.</u> Many good educators shy away from computers because they don't have the programming skills necessary to make the machines do what they want. Now that you have completed all of the background work, it's a simple matter to tell an actual programmer what you want to see on the screen.

An extremely effective way of doing this is through the use of a storyboard. The storyboard is the bread and butter of Hollywood when it comes to shooting a movie. In our case, it's a number of squares which represent the consecutive screens of the computer. Your job is to draw in each of the squares what you want to see and include to the side any notes concerning actions. This system makes communication between you and your programmer clear and exact which will eliminate many (but not all) frustrations.

The same is true for your other educational materials. You may be great at instructional design (and I wish more people were), but that doesn't mean you can write. If you know what you want, you might want to go through the same process when explaining to a writer what you want. This is the world of specialization, so specialize.

This is a step that most developers forget. "It's too costly," "It takes too much time," or "We don't have anyone to test it on." All of these excuses are falacies. There is no excuse for releasing educational systems that are defective.

It's never too costly to ensure the quality of a product. If it's good and produces the desired results, people will hear about it. Increased sales will more than make up for any expenditures incurred in testing.

The same is true when it comes to time. Debugging and field testing time should be planned for in your development time. There's nothing that will kill the interest in a program faster than execution bugs. These must be ferreted out before the public finds them. I know of one software house that takes this so seriously that they unleash a group of kids on their products before releasing them and offer a reward of $5 to whomever finds a bug.

Always take the time to test your system before your start distributing it. You can always use your own children, neighborhood kids or try on the population in nearby schools. If you promise to give them a copy of the finished product, schools are usually receptive to trying out new products.

REEVALUATE YOUR PROGRAM

After you've field tested your system, reevaluate the results. Did the students actually learn what you had hoped? Were there any difficulties in the running of the project? How can you make it more effective?

The best way to answer these questions is to interview the teachers and students involved. They're the ones who use it and they're the ones who know best what worked and what didn't.

Another way of evaluating your project is to watch the kids working on it. Only through observation can you see how the kids interact with your system. Only through observation will you be able to make it more ergonomically sound.

GO TO IT

Now that you know the fundamentals of producing an educational system, go to the library and find out the specifics. There are a great number of books out on instructional design and my brief synopsis is by no means complete. Once you feel satisfied that you know enough to attack the project of building an educational system, <u>do it.</u> There is no greater teacher than experience and only through doing it will you be able to find out what it's all about.