SHARON DUGDALE

# What's the Student Doing?—A Crucial Aspect of Instructional Design

The increasing availability of low-cost microcomputers for classroom use has led to an outpouring of instructional materials using this relatively new medium. However, in creating or choosing computerized instructional materials for classroom use, we too often overlook a crucially important question: what is the student actually doing in this particular exercise? The answer to this question is quite independent of the use of graphics, the cleverness of the format, and other factors that influence our opinions of materials. However, it may well be the most important consideration in identifying educationally effective courseware.

One way to address this question is to carefully disregard everything about the program that the student can easily ignore—that is, strip the program back to the actions the student must make, and the information to which the student must attend.

The following examples illustrate this point. The descriptions are as they might appear in the marketing literature.

Our first example is a well-known program that has appeared on several microcomputers over the past few years. The description here is of the original version[1] and may differ in some ways from the various versions that have since been programmed for other hardware.

The thought and activity engaged in by the student may well be the most important consideration in choosing or creating educationally effective instructional materials. Whether we like it or not, we may as well admit that most precollege mathematics students have learned to approach instructional materials with the question "What am I supposed to *do*?" rather than "What am I supposed to *learn*?" Once the student knows what to do, there is no need to seek further information from the display. In the second example, the explicit instructions ("Type a number between 1 and 10") could be removed, but then the student would likely be confused until finding out what to do. Once the student understands, the reaction is, "Oh, I see. All you have to do is put in numbers. Why didn't they just say that?"

In short, if the mathematics (or whatever subject matter) is not inherent in what the student is *do*ing, then it is not likely to be inherent in what the student is *think*ing. ■

## 1. Darts

This program allows the student to explore the placement of rational numbers on the number line. Balloons are tied to a number line, and the student shoots darts at the balloons by estimating their positions on the line (Figure 1). Fractions, decimals, mixed numbers, and expressions using operations are all acceptable inputs.

When the student inputs a number, a dart flies across the screen and lodges in the number line at the specified location, popping a balloon if it hits one. The student's input is shown beside the dart (Figure 2).

The student's performance is continually monitored, and the difficulty of the task is adjusted to a challenging but comfortable level for each student. The easiest levels have larger balloons and more integers on the number line (hence a shorter space between integers). More difficult levels have smaller balloons and fewer integers on the number line.

As in the other example, the student is to type a number. Again, it is true that any number within the given interval will "do," in that it will fire a dart. However, in order to hit the balloons, the student clearly must attend to, and use, the mathematics involved. The student simply cannot play the game without actively thinking about the placement of rational numbers on the number line.

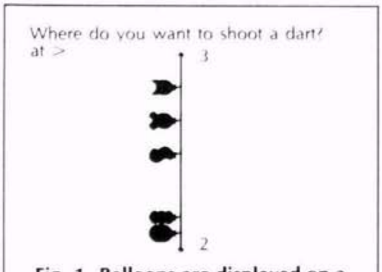


**Fig. 1. Balloons are displayed on a number line. The student is ready to shoot a dart by typing a number to specify a position on the number line.**

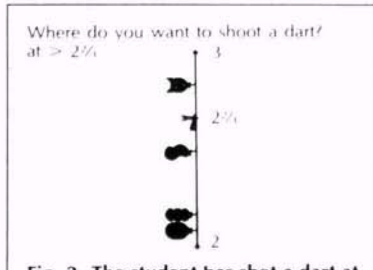


**Fig. 2. The student has shot a dart at $2\frac{2}{3}$ and hit a balloon. The student's input is shown beside the dart. The game continues until all of the balloons have been hit.**

**Before selecting a computer program, check to see whether it requires students to think.**

## 2. Problem Solving: An Experiment with Distance, Rate, Time Problems

This is not an exact description of any particular program for microcomputers. However, it is quite similar to some existing materials, from which the essential characteristics are taken.

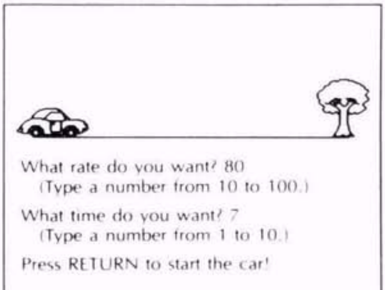


**Fig. 1. The student choses a rate and time and is ready to start the car.**

This simulation teaches distance, rate, time problems by giving the student actual experience with the physical motion involved. The student controls the rate of speed and the time of travel for a car on the screen.

After specifying the rate and time, the student presses RETURN to start the car, as in Figure 1. As the car moves across the screen, the distance traveled is shown under the car's path. To add realism to the experiment, the distance is kept dynamically, as if the student were watching the odometer of the car, shown in Figure 2. The car stops at the appropriate distance, and the student records the distance traveled, in Figure 3. The program ends after ten problems.

This program description has all the right words—"problem solving," "simulation," "experiment," and so on. The student has lots of control. The instructions on the screen are clear and leave almost no chance of failure or misunderstanding. The graphics and animation are not only cute, they are relevant to the problem as well.

But let's look a little deeper and see what the student is actually involved in. A quick analysis reveals that the student must perform the following tasks:

1. Pick a number from 10 to 100.
2. Pick a number from 1 to 10.
3. Watch the car and counter go. (This is, of course, optional, but students are likely to do it.)
4. Copy the number from the counter.

The information to which the student must attend in order to do these tasks is simply the computer's request to do them.

It seems reasonable to ask whether repeating this sequence ten times will really further a student's understanding of or facility with anything but picking and copying numbers. Clearly, a student can complete the program successfully with no notion of its instructional goal, and further, with no notion that there even was a goal beyond entering two numbers and copying a third. Even if the student notices that the counter always stops at the product of the two chosen numbers, there is little reason to believe the experience is likely to be related to the goals in the program description, or that the student's thoughts center on anything that could rightfully be called "problem solving."

Of course, this program could be useful if the classroom teacher were to sit with the student and ask appropriate questions to focus the student's attention beyond the trivial clerical tasks demanded by the program. Although there may be a use for this mode of instruction, I doubt that most computerized materials are bought or created with the intention of requiring close teacher attention during use. ■

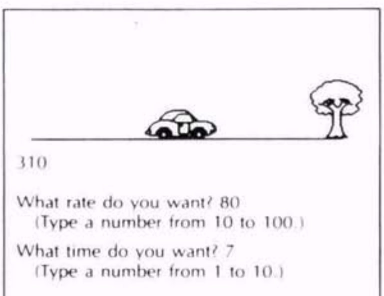


**Fig. 2. As the car moves, the distance is recorded on the counter.**

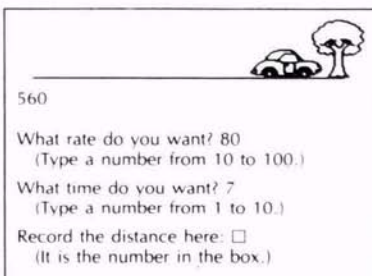


**Fig. 3. The car has stopped. The student is asked to record the distance.**

[1]Created in 1973 by Sharon Dugdale on the PLATO CBE system, with funding provided by the National Science Foundation (US NSF C-723).

*Sharon Dugdale is Senior Specialist in Automated Education and Director of the PLATO Mathematics Project, Computer-based Education Research Laboratory, University of Illinois at Urbana-Champaign.*